T90-11094

JSC 24626

NASA

National Aeronautics and
Space Administration

AUTONOMOUS RENDEZVOUS AND DOCKING CONFERENCE

Volume II

Lyndon B. Johnson Space Center
Houston, Texas

August 15-16, 1990

Sponsored by

NASA Office of Space Flight

NASA Office of Aeronautics, Exploration and Technology

NASA Space Servicing Systems Project Office

AUTONOMOUS RENDEZVOUS AND DOCKING CONFERENCE

August 15-16, 1990

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

PREFACE

This document consists of the presentations submitted at the Autonomous Rendezvous and Docking (ARD) Conference. The document contains three volumes:

VOLUME I	ARD Hardware Technology
VOLUME II	ARD Software Technology
VOLUME III	ARD Operations

Information contained herein should not be construed as being the official NASA position. Responsibility for content and technical accuracy lies with each respective author.

The ARD Conference was sponsored by NASA Office of Space Flight, NASA Office of Aeronautics, Exploration and Technology, and NASA Space Servicing Systems Project Office.

James S. Moore

James S. Moore
Manager of NASA Space Servicing Systems Project Office
NASA-JSC
Code IA 141

AUTONOMOUS RENDEZVOUS AND DOCKING CONFERENCE COMMITTEE
August 15-16, 1990

Conference Chairman	James S. Moore Manager of Space Servicing Systems Project Office NASA-Johnson Space Center
Sponsor Representatives	George M. Levin Chief of Advanced Space Systems NASA Office of Space Flight
	John C. Mankins Acting Manager of Pathfinder Program NASA Office of Aeronautics, Exploration and Technology
Program Chairman	Robert G. Richmond Lockheed Engineering & Sciences Company
Technical Session Chairmen	
Session I ARD Hardware Technology	Dr. Neville Marzwell NASA-Jet Propulsion Laboratory
Session II ARD Software Technology	Aldo J. Bordano NASA-Johnson Space Center
Session III ARD Operations	Don J. Pearson NASA-Johnson Space Center
Conference Coordinator	Lana R. Arnold Lockheed Engineering & Sciences Company

PAGE iv INTENTIONALLY BLANK

INTRODUCTION

Autonomous Rendezvous and Docking (ARD) will be a requirement for future space programs. Clear examples include satellite servicing, repair, recovery, and reboost in the near term, and the longer range lunar and planetary exploration programs. Indeed, ARD will permit more aggressive unmanned space activities, while providing a valuable operational capability for manned missions. The purpose of this Conference is to identify the technologies required for an on-orbit demonstration of ARD, assess the maturity of those technologies, and provide the necessary insight for a quality assessment of programmatic management, technical, schedule, and cost risks.

James S. Moore
ARD Conference Chairman

PROCEEDINGS CONTENTS

PAGE viii INTENTIONALLY BLANK

PROCEEDINGS CONTENTS (CONTINUED)

CONTENTS-VOLUME II

Session II ARD SOFTWARE TECHNOLOGY

Aldo J. Bordano
NASA-Johnson Space Center
Technical Session Chairman

AUTONOMOUS RENDEZVOUS AND DOCKING SOFTWARE TECHNOLOGY

SESSION II

SUMMARY

Aldo J. Bordano

NASA - Johnson Space Center

ARD SOFTWARE TECHNOLOGY

- CANDIDATE GN&C ALGORITHMS HAVE BEEN DEVELOPED AND SOME PRELIMINARY TESTING HAS OCCURRED
- NO SIGNIFICANT TECHNICAL ISSUES EXIST
- DURING PHASE B, CANDIDATE ALGORITHMS SHOULD BE EVALUATED AND AN APPROACH ADOPTED FOR IMPLEMENTATION DURING PHASE C/D
- NEURAL NETWORK SOFTWARE HAS SOME INTERESTING POTENTIAL BUT PROOF OF CONCEPT IS STILL ABSENT FOR AR&D APPLICATION
- LAMBERT AND KEPLER ALGORITHMS HAVE APPLICATION FOR GUIDANCE
- AN LQG/LTR APPROACH FOR CONTROL HAS BEEN DEVELOPED AND IS A CANDIDATE
- HERMES AND COLUMBUS/CNES HAVE ACCOMPLISHED SIGNIFICANT GN&C DEVELOPMENT AND TESTING

ARD SOFTWARE SUMMARY
(cont'd)

- A NUMBER OF SIGNIFICANT TEST BEDS EXIST IN SUPPORT OF AR&D
  - CNES/HERMES AND COLUMBUS (HW/SW, DOCKING. BERTHING MECHANISMS, ETC.)
  - MARTIN MARIETTA (REAL TIME AND NON REAL TIME, HW/SW, MAN-IN-THE-LOOP, MANIPULATORS, SCALE MODELS, ETC.)
  - LIN COM (INTEGRATED SOFTWARE TEST BED)
  - MSFC AIR BEARING TABLE
- A PARTIAL LIST OF S/W OPEN ITEMS INCLUDE:
  - MAN/GROUND INTERVENTION/SUPERVISION S/W
  - GPS UTILIZATION (YES/NO)
  - GN&C PERFORMANCE REQUIREMENTS
  - TEST BED UTILIZATION
- A GOOD PHASE B IS NEEDED TO BETTER FOCUS THE END-TO-END S/W REQUIREMENTS AND CONCEPT SELECTION

ARD SOFTWARE TECHNOLOGY

ABSTRACTS

Universal Lambert and Kepler Algorithms for Autonomous Rendezvous

Allan R. Klumpp
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA. 91109

New Lambert and Kepler algorithms, designed to be the core of an autonomous rendezvous guidance system aboard a flight computer, have been developed. Applications include robotic and piloted missions to the moon and planets. Flight software must be compact, fast, and totally reliable. Although high accuracy is not essential for flight, in double precision these algorithms are accurate to at least 14 places almost everywhere. Both are universal; they apply to elliptic, parabolic, hyperbolic, and even rectilinear trajectories. The algorithms are improvements to those published by Richard Battin in his 1897 text. The Lambert algorithm provides the best of several solutions for initial and terminal velocity vectors (rather than the test's semimajor axis and semilatus rectum), it solves for the plane of the trajectory when the plane is indeterminate from the initial and terminal position vectors, and it introduces new convergence criteria that achieve the computer's full potential accuracy in the minimum number of iterations. The Kepler algorithm reorganizes the test's equations and solves them by a binary search, thereby guaranteeing convergence (the text's algorithm does not converge for a class of realizable trajectories).

A Linear Quadratic Gaussian With Loop Transfer Recovery Proximity Operations Autopilot for Spacecraft

George T. Chen
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

An automatic control scheme for spacecraft proximity operations is presented. The controller is capable of holding the vehicle at a prescribed location relative to a target, or maneuvering it to a different relative position using straight line-of-sight translations. The autopilot uses a feed-forward loop to initiate and terminate maneuvers, and for operations at non-equilibrium set points. A multivariable feedback loop facilitates precise position and velocity control in the presence of sensor noise. The feedback loop is formulated using the Linear Quadratic Gaussian (LQG) with Loop Transfer Recovery (LTR) design procedure. Linear models of spacecraft dynamics, adapted from the Clohessey-Wiltshire Equations, are augmented with loop shaping techniques that are applied to design a target feedback loop. The loop transfer recovery procedure is used to recover the frequency-domain properties of the target feedback loop. The resulting compensator is integrated into an autopilot which is tested in a high-fidelity Space Shuttle simulator. The autopilot performance is evaluated for a variety of proximity operations tasks envisioned for future Shuttle flights.

The results of high-fidelity, nonlinear simulations has verified the applicability of LQG/LTR compensation to feedback control of spacecraft translational dynamics. Even with realistic levels of sensor noise and thruster jet granularity, the autopilot has demonstrated the capability of guiding the vehicle through typical proximity operations tasks, such as v-bar and r-bar approaches, with high precision while maintaining fuel consumption at an acceptable level.

Rendezvous Simulation and Error Analysis

Nick G. Smith
Martin Marietta
P. O. Box 179
Denver, Co. 80201

Autonomous rendezvous and docking (ARD) is a critical space technology with a variety of applications. The Mars Rover/Sample Return (MRSR) mission is an example of the significant benefit that can be derived by incorporating ARD technology. Preliminary studies have shown MRSR launch mass sensitivities as high as 100 percent, if ARD is not utilized. Martin Marietta Astronautics Group (MMAG) has been developing ARD technical capability in support of the MRSR Delivery and Return study, contracted by the NASA Johnson Space Center. MMAG has also been developing ARD software under independent research and development studies.

One of the ARD tools developed is the Simulation and Error Analysis of Rendezvous Trajectories (SEART) program. SEART has been applied to the MRSR mission to determine the key parameters that drive rendezvous performance. MRSR rendezvous has been defined as the transfer of the active vehicle from an orbit co-elliptic with the target vehicle to a position one kilometer in front of the target vehicle. This maneuver includes the transfer phase initialization (TPI) burn, the transfer phase finalization (TPF) burn, and any required mid-course correction burns. Rendezvous performance can be characterized by the ideal delta velocity required and by the final errors at the end of the rendezvous phase. Initial navigation state errors and sensor measurement errors have been found to be the major rendezvous performance drivers. Performance sensitivities to these errors have been quantified through Monte Carlo analyses.

The presentation describes the rendezvous simulation and error analysis software tool (SEART) and discusses rendezvous performance results for the MRSR mission study.

Autonomous Orbital Operations Software Testbed

Daniel C. Bochsler
LinCom Corporation
1020 Bay Area Blvd Ste. 200
Houston, TX. 77058

The objective of this project is to provide an integrated software testbed for design of autonomous space vehicle systems and spacecraft as well as for mission analysis and design. Development activities are building upon initial testbed work which was started in late 1988 which emphasized architectural and component functions. Advance planning and technology utilization are oriented toward the goal of a testbed providing a flexible framework which will support early and later stages of new as well as existing spacecraft hardware, software, and operations development. A key strategy is to minimize new development efforts by leveraging and utilizing results and methods from other technology development efforts; the integrated nature of an autonomous operations testbed involving many technologies and methods will require significant system engineering and integration efforts. Particular emphasis will be placed on multiple vehicle interoperability in detailed operational scenarios, including the complex physical and operational interaction of manipulator systems with manned and unmanned vehicles. The testbed will accommodate multiple vehicles in a modular "plug-in" fashion in order to permit substitution of alternative vehicles from a library of vehicles maintained for the testbed. The vehicles themselves will be built up from modular plug in modules as well as for individual subsystems, sensors, efforts, dynamics, etc. This project is being sponsored through a Code M RTOP and managed through the Software Technology Branch at JSC.

Satellite Servicer System End-to-End Simulation

John A. Cuseo
Space Operations Simulation Laboratory
Martin Marietta Astronautics Group
P. O. Box 179
Denver, Co. 80201

An end-to-end satellite servicer system simulation has been developed. This high-fidelity simulation operates in realtime, with hardware-in-the-loop and covers all aspects of a typical satellite servicing scenario including: element deployment and retrieval from the Shuttle, proximity operations, autonomous rendezvous and docking, orbital replacement unit (ORU) exchange, and fluid resupply. This type of simulation requires a level of development typical of a flight system, thus giving high confidence in trade studies of design concepts for rendezvous and docking systems.

The core of the simulation capability is a comprehensive, realtime, distributed computing architecture linked by an Encore (Gould) 32/9750 simulation controller. The Encore contains the high-fidelity modelling of spacecraft systems, multi-body dynamics, guidance and control systems, and the orbital environment. The Encore provides state vectors to a large amplitude (400 m^3 maneuvering volume), 6-degree-of-freedom moving base carriage (MBC), and target gimbals (1 and 3-axis) to simulate the rotational and translational state of the chase vehicle with respect to the target. The MBC has a payload capability of 320 kg to support installation of representative docking hardware and sensor systems. Successful rendezvous and docking sequences have been simulated using video sensors and an image processing system, the Geometric Arithmetic Parallel Processor (GAPP). The GAPP is used to locate and center four known points on the target vehicle whose image plane coordinates snd are then used to determine relative position and orientation as input into the guidance and control system. A Laser Docking Sensor (LDS) software model was also developed and integrated into the simulation to study integration issues and error sensitivities.

Both ground control and on-orbit workstations are integrated into the simulation for man-in-the-loop studies related to supervised autonomous rendezvous and docking systems. These workstations incorporate state-of-the-art hardware and software, such as rapid prototyping systems, touchscreen and sensor data overlays, integrated speech synthesis and recognition, and numerous hand controller options. With these workstations, such key user interface issues as the allocation of control functions in supervised autonomous systems can be studied in the context of realtime, high-fidelity mission tasks.

Hermes and Columbus Rendezvous Control System

Christian M. Pauvert, Jean-Michel Pairot
MATRA ESPACE, Toulouse, France
Dr. W. Fehse, A. Tobias
European Space Agency, Noordwijk The Netherlands

The design of a reference RV control system concept applicable to the missions between Hermes, the Columbus free-flyer, and the U.S. Space Station is presented. This concept, elaborated in the frame of the ESA RVD Proof-of-Concept Programme, covers all the phases from the homing until docking or berthing. Retreat and emergency cases, in accordance with the mission scenarios established on the basis of project requirements and baselines, are also presented and discussed. The presentation describes the RV control algorithms, including the Guidance, Navigation and Control, the mission management, the vehicle configuration management and the FDIR. Emphasis is placed on the navigation using the absolute and relative GPS measurements processed through a Kalman filter, and the coupled processing of the RV sensor (CCD camera) attitude and position measurements. The interaction between navigation and control during the last meters before contact is discussed.

Performance results of simulations of the last twenty meters, linked to the preliminary specifications of the RV sensor, are shown.

The second portion of the presentation is dedicated to the test activities of the RV control system. A prototype of the software is being developed covering all the mission phases for both non-realtime closed-loop tests, and for realtime closed loop tests for the last twenty meter approach (utilizing a representative realtime processor).

ARD SOFTWARE TECHNOLOGY

PRESENTATIONS

UNIVERSAL LAMBERT AND KEPLER ALGORITHMS FOR AUTONOMOUS RENDEZVOUS

ALLAN KLUMPP
JET PROPULSION LABORATORY
PASADENA, CAL 91109

AUTONOMOUS RENDEZVOUS AND DOCKING CONFERENCE
JOHNSON SPACE CENTER
HOUSTON, TEXAS
1990 AUGUST 15-16

SUMMARY

New Lambert and Kepler algorithms, designed to be the core of an autonomous rendezvous guidance system aboard a flight computer, have been developed. Applications include robotic and piloted missions to the moon and planets. Flight software must be compact, fast, and totally reliable. Although high accuracy is not essential for flight, in double precision these algorithms are accurate to at least 14 places almost everywhere. Both are universal; they apply to elliptic, parabolic, hyperbolic, and even rectilinear trajectories. The algorithms are improvements to those published by Richard Battin in his 1987 text. The Lambert algorithm solves for the plane of the trajectory even when the plane is indeterminate from the initial and terminal position vectors, it introduces new convergence criteria that achieve the computer's full potential accuracy in the minimum number of iterations, and it provides the solution for initial and terminal velocity vectors that ranks best among several candidates tested. Unlike the text, the Lambert algorithm computes the velocities, not the semimajor axis and semilatus rectum. The Kepler algorithm reorganizes the text's equations and solves them by a binary search, thereby guaranteeing convergence (the text's algorithm does not converge for a class of realizable trajectories).

CONTENTS

UPPER PAGE OF HANDOUT

VUGRAPHS AND LOWER PAGE OF HANDOUT

BACKGROUND

Gauss first solved Lambert's problem 1801 September, and used the solution that October to determine the orbit of Ceres from a 3^{o} arc traversed in 41 days. Gauss published solutions to both Lambert's and Kepler's problems in 1809 in his *Theoria Motus*. Although effective for his applications, Gauss' algorithms fell far short of the needs of contemporary astrodynamics. The Lambert algorithm was singular for 180^{o} transfers. It was often slow to converge, and it failed to converge for a wide spectrum of hyperbolic cases. The Kepler algorithm was designed for near-parabolic orbits, and it was useful only for short arcs reckoned from periapse.

Lancaster and Blanchard[1] presented a new Lambert algorithm which is exact and valid for multiple-orbit transfers. Although its solution for the velocity vector is singular for transfer angles that are a multiple of 180^{o} (including 0^{o}), R. H. Gooding of the Royal Aerospace Establishment has apparently overcome this limitation, and has had a manuscript describing his algorithm accepted for publication in *Celestial Mechanics*.

Richard Battin and his thesis students at the Massachusetts Institute of Technology have recently made great strides in extending the Gauss algorithms to meet the needs of contemporary astrodynamics. Battin and Vaughan[2,3] improved the Lambert algorithm so that it converges for virtually all realizable trajectories. It is singular only for 360^{o} equiradius transfers, and it fails to converge only for trajectories microscopically close to the singularity. Battin and Loechler[4] extended the Lambert algorithm to handle multiple-orbit transfers, but this extension is not implemented here because it substantially complicates the algorithm. Battin and Fill[5,6] extended the Kepler algorithm to trajectories not necessarily reckoned from periapse in which the eccentricity and arc length are arbitrary. Improved versions of these algorithms were published in Battin's 1987 book,[7] henceforth referred to as the text.

The text algorithms have shortcomings, some of which were discovered only by testing the algorithms on a spectrum of trajectories spanning the multi-dimensional independent parameter space. The Lambert algorithm solves only for the semimajor axis and semilatus rectum; no solution for the initial and terminal velocity vectors is included with the algorithm, although the equations needed for deriving an excellent solution appear elsewhere in the text. The semimajor axis and semilatus rectum can also be used for deriving a solution, but it is very inaccurate for some common cases. The convergence criteria unnecessarily limit the algorithm's accuracy. In the vicinity of the singularity, the algorithm can generate erroneous results without indication of error. The text's Kepler algorithm fails to converge for a group of elliptic trajectories with negative initial flight path angle (the angle between the velocity vector and the local horizontal, positive upward). Convergence can fail for any value, even infinitesimal, of generalized anomaly (the algorithm's dimensionless representation of the transfer angle).

The new algorithms correct the above shortcomings while they inherit almost all of their virtues and all of their important equations from the algorithms of the text.

TWO FUNDAMENTAL PROBLEMS IN ASTRODYNAMICS

Lambert's Problem

- Find the two-body trajectory that connects two given position vectors in a given transfer time
- Inputs: R_1, R_2, transfer time
- Outputs: V_1, V_2

Kepler's Problem

- Find the two-body trajectory passing through a given initial state (position and velocity)
- Inputs: R_1, V_1, transfer time
- Outputs: R_2, V_2

DEFINING THE PROBLEMS

Lambert's Problem

Lambert's problem is defined by the initial and terminal position vectors, the transfer time, and the gravitational constant of the central body (considered a point mass). The problem is to find the initial (or terminal) velocity vector that produces the transfer. (Either velocity vector is sufficient; this algorithm solves for both.)

Lambert's problem is two dimensional. The two position vectors define the plane of the trajectory. The entire trajectory must be confined to this plane since the central body is a point mass. Thus two components of the initial velocity vector constitute a solution; the component normal to the plane of the trajectory must be zero.

Kepler's Problem

Kepler's problem is defined by the initial state, transfer time, and the gravitational constant of the central body (considered a point mass). The problem is to find the terminal state.

Kepler's problem is one dimensional because the trajectory is defined by the given initial state. The problem amounts to finding the point along the trajectory that corresponds to the given transfer time.

APPLICATIONS

THEORETICAL ASTRONOMY

- o Use Lambert to determine a two-body planetary orbit from two observations of position
- o Use Kepler to determine a two-body planetary orbit from one known state

RENDEZVOUS GUIDANCE

- o Use Lambert to find required velocity, given the position of the active spacecraft at maneuver time and the position of the target spacecraft at rendezvous time
- o Use Kepler to find projected miss vector, given the current state of the active spacecraft and the position of the target spacecraft at rendezvous time

FEATURES OF THE ALGORITHMS

The algorithms are sufficiently reliable, compact, and fast to be used on flight computers. They have the accuracy required for theoretical astrodynamics. They are applicable to every type of trajectory, elliptic, parabolic, hyperbolic, and even rectilinear. Trajectories about bodies in our solar system, computed in MKS units, can be flown to within the wavelength of light of the center of the attracting body (represented as a point mass). On VAX computers executing in G_floating and IBM mainframes executing in double precision, the Lambert algorithm can compute the orbit of a sun-grazing comet whose period is 3777 years, using two observations symmetrical about apoapse but separated by 359.999_992°. While apoapse radius is 485 AU (over 12 times the radius of Pluto's orbit), at the second observation the comet has returned to within 10,158 km of its starting position. Determining the orbit from two observations with the same geometry but 360° minus the same transfer angle is trivial.

FEATURES OF THE ALGORITHMS

- Sufficiently compact and fast for flight computers
- Universal: applicable to any type of trajectory; elliptic, parabolic, hyperbolic, rectilinear

LAMBERT

- Trajectories (rectilinear) can pass thru the center of the attracting body
- Singular for 360° equiradius transfers, but equiradius transfers of 359.999_992° can be computed

KEPLER

- Numerical overflow at center of attracting body, but, using MKS units, trajectories can pass within the wavelength of light of the center of a planet
- No singularities; 360° transfers can be computed

LIMITATIONS OF THE ALGORITHMS

There is no claim that the algorithms cannot be further improved. Although the Lambert algorithm normally converges in six iterations (31 milliseconds on a DEC VAXstation 3100), it takes up to 660 iterations for near-360° minimum-energy orbits. As mentioned, it does not handle multiple-orbit transfers. The Kepler algorithm normally converges in about 100 milliseconds. It uses a binary search; the number of iterations is typically the number of bits in the mantissa, often around 53. A limited number of additional iterations is required when the transfer angle is extremely small.

LIMITATIONS OF THE ALGORITHMS

LAMBERT

- o Handles only single-orbit transfers, although it can be modified to handle multiple-orbit transfers at a substantial cost in code

KEPLER

- o The number of iterations is typically the number of bits in the mantissa, often around 53, making Kepler not as fast as Lambert

LAMBERT IMPLEMENTATION

The orbit normal is needed for computing the initial and terminal velocity vectors. The obvious solution for the normal is the cross product of the initial and terminal position vectors. But for transfers in the vicinity of 0^{o}, 180^{o}, and 360^{o}, this solution becomes indeterminate. An alternate solution is the cross product of the initial position vector and the premaneuver velocity vector.

Both solutions are computed. Instead of switching from one to the other at predetermined arbitrary transfer angles, the algorithm mixes the two solutions. Therefore the transition from one solution to the other is a continuous function of transfer angle. Step changes in the direction of the orbit normal are thereby avoided. A mixing coefficient is chosen to make the solution based on position vectors dominate whenever it is well conditioned.

The algorithm uses successive substitutions for iterating toward a solution. The iterative process is terminated (the algorithm is converged) when the solution repeats any one of the last n solutions (n = 12). This convergence criterion offers the following benefits compared to quantitative criteria:

- The full potential accuracy of the computer is achieved in the minimum number of iterations.
- Convergence accuracy is automatically adjusted for the difficulty of the trajectory.
- Machine-dependent constants are avoided (although two machine-dependent constants are still used for other purposes). This enhances portability of the algorithm.

The best solution for the velocity vectors was chosen from among several candidates tested. All other solutions were significantly less accurate for one or more cases in a suite of over 100 test cases.

LAMBERT IMPLEMENTATION

- Orbit normal computed as cross product of initial and terminal position vectors mixed with cross product of initial position and premaneuver velocity vectors
- Mixing coefficient for orbit normal chosen to make initial and terminal positions dominate except when they are near collinearity
- Iterative process is successive substitutions, valid for all cases
- Iterative process is terminated when a solution is repeated; no numerical criteria are used
- Best solution for velocity vectors was chosen from among several candidates tested

KEPLER IMPLEMENTATION

Kepler's problem can be reduced to a sequence of equations which can be evaluated in closed form to yield a value for normalized transfer time in terms of a generalized anomaly peculiar to the algorithm. This sequence of equations is called "Kepler's equation". Kepler's problem is then to invert Kepler's equation, that is, to solve for the generalized anomaly given the normalized transfer time. Using the generalized anomaly, the terminal state is computed in closed form.

One of the equations in the sequence is expressed as a truncated infinite series. This series is valid only over a limited range of values of the generalized anomaly. Therefore, following Gauss and Battin, the magnitude of the generalized anomaly is limited to 3/10. This limit corresponds to an arc of 64^{o} along a circular orbit, but it corresponds to an infinite transfer on a parabolic orbit where the generalized anomaly is zero.

Because of this limit, the algorithm computes elliptic and parabolic transfers in any number of fixed steps of maximum size, followed by a single variable step. For the fixed steps, Kepler's equation is simply evaluated, not solved. For the variable step, the algorithm solves Kepler's equation by a binary search.

A binary search is used because a plot of Kepler's equation can have a zero-slope region and can be multiple valued. The search is terminated when Kepler's equation is solved exactly; there is no quantitative convergence criterion. The benefits of exact convergence are:

- o The full potential accuracy of the computer is achieved in the minimum number of iterations.
- o The algorithm is free from variables whose values depend upon word length or other characteristics of the language or the machine. Therefore, it should be completely portable without implementation-dependent adjustments.

Although a binary search requires many iterations, processing time should be not be a problem on a flight computer (see "Performance").

In the parabolic case, the algorithm solves Kepler's equation in a single iteration, using the method described in the text. The parabolic case entails finding one real root of a cubic equation. Ref. 8 proves that, for parabolic trajectories, there is exactly one real root, except in two cases. The first exceptional case is when the transfer time is zero. The second is when the terminal position coincides with the central body, considered to be a point mass. Although neither of these cases has much practical significance, the proper root is easily found in both.

Unlike the text's algorithm, the algorithm described here handles any transfer time, positive or negative. If the given transfer time is negative, the algorithm uses the negative of the given transfer time and the negative of the initial velocity vector, and returns the negative of the velocity vector it computes.

KEPLER IMPLEMENTATION

- For elliptic and hyperbolic cases, computes the transfer in any number of fixed steps of maximum size (64° on a circular orbit), followed by a single variable step
  - On fixed steps, evaluates Kepler's equation for normalized transfer time in terms of generalized anomaly
  - On variable step, solves Kepler's equation for generalized anomaly in terms of normalized transfer time
  - Solves Kepler's equation by a binary search; convergence guaranteed
- For parabolic case, text algorithm is used
  - Solves Kepler's equation in a single iteration
- Seams on elliptic-parabolic and parabolic-hyperbolic boundaries are invisible
- Negative transfer time handled by reversing signs of given time, given velocity, and computed velocity

PERFORMANCE

Ref. 9 shows that both algorithms provide correct solutions for all realistic trajectories. In fact, the limits of the algorithms lie far beyond what is demanded by realistic trajectories.

Accuracy of the algorithms is defined as the minimum number of decimal places of agreement between the terminal position vector input to the Lambert algorithm and that computed by the Kepler algorithm. The state input to Kepler consists of the initial position vector input to Lambert and the initial velocity vector computed by Lambert.

Accuracy is at least 14 places everywhere, and usually 15 places, except for the following cases:

- For very slow transfers, e.g., 30° transfer angle with transfer time 1000 times that for a parabolic trajectory, the accuracy drops to 11 places.
- For orbits in which the orbit plane is nearly indeterminate from the initial and terminal position vectors (e.g., transfer angles close to 0°, 180°, or 360°), the accuracy may drop to around 11 places.
- For near-360° equiradius transfers, the accuracy degrades, particularly as the orbit becomes rectilinear and the periapse radius approaches zero. For example, for a trajectory beginning well above Mars' atmosphere and penetrating Mars to an 8.9-nanometer periapse, the accuracy drops to two places.

The accuracy of either an elliptic or a hyperbolic trajectory bordering the parabolic boundary is 15 places.

The Lambert algorithm converges in seven or fewer iterations everywhere except in a region surrounding its singularity. In this region, the number of iterations increases to a maximum of 660. Ref. 9 provides the details.

The Kepler algorithm takes typically 53 iterations to converge (VAX G floating and IBM Mainframe double precision). The number of iterations increases when the transfer angle is less than about $7E\text{-}15^{\circ}$.

The algorithms are fast enough for most flight applications using modern flight computers. The following results are for an otherwise unloaded VAXstation 3100 model 30:

- A suite of 22 Lambert cases searching for the case that requires the maximum number of iterations executes in 76 seconds. Of the 22 cases, 16 take 659 iterations and 6 take 660. This is 5.24 milliseconds per iteration or about 31 milliseconds for the normal case requiring 6 iterations.
- A suite of 144 Kepler test cases, with the number of iterations ranging from 52 to 57, executes in 15 seconds. This is about 104 milliseconds per case.

PERFORMANCE

- o Both algorithms provide correct solutions for all realistic trajectories
- o Accuracy is at least 14 places everywhere, and usually 15 places, except in extreme cases
- o The accuracy of either an elliptic or a hyperbolic trajectory bordering the parabolic boundary is 15 places
- o The Lambert algorithm converges in seven iterations except in a region close to its singularity where the number of iterations increases to a maximum of 660
- o The Kepler algorithm converges typically in 53 iterations (VAX G_floating and IBM Mainframe double precision); more iterations are needed when the transfer angle is less than about 7E-15^{o}
- o Lambert execution time on a VAXstation 3100 model 30 is typically 31 ms
- o Kepler execution time on the same machine is typically 104 ms

REFERENCES

1. E. R. Lancaster and R. C. Blanchard, "A Unified Form of Lambert's Theorem", NASA TN D 5368, Goddard Space Flight Center, Greenbelt, MD, 1969 September.

2. R. M. Vaughan, "An Improvement of Gauss' Method for Solving Lambert's Problem", Master of Science Thesis, Massachusetts Institute of Technology, 1983 May.

3. R. H. Battin and R. M. Vaughan, "An Elegant Lambert Algorithm", *Journal of Guidance, Control, and Dynamics*, Vol 7 Number 6, 1984 November-December.

4. L. A. Loechler, "An Elegant Lambert Algorithm for Multiple Revolution Orbits", Master of Science Thesis, Massachusetts Institute of Technology, 1988 May.

5. T. J. Fill, "Extension of Gauss' Method for the Solution of Kepler's Equation", Master of Science Thesis, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MAS 02139, 1976 May.

6. R. H. Battin and T. J. Fill, "Extension of Gauss' Method for the Solution of Kepler's Equation", *Journal of Guidance and Control*, 1979 May-June.

7. R. H. Battin, "An Introduction to the Mathematics and Methods of Astrodynamics", American Institute of Aeronautics and Astronautics, Inc., 1633 Broadway, New York, N.Y. 10019, 1987.

8. A. R. Klumpp, "Universal Lambert and Kepler Procedures in Ada", JPL D-7470, (internal document), 1990 June.

9. A. R. Klumpp, "Universal Lambert and Kepler Algorithms for Autonomous Rendezvous", AIAA/AAS Astrodynamics Conference, Portland, Oregon, 1990 August 20-22.

10. A. R. Klumpp, "An Autopsy of the Extended Gauss Kepler Algorithm", IOM 314.1-0307-ARK, (internal document), 1989 August 23.

Version of 90 08 13

JPL

A Linear Quadratic Gaussian With Loop Transfer Recovery Proximity Operations Autopilot for Spacecraft

George Chen

Guidance & Control Section
Jet Propulsion Laboratory

The work presented here was made possible by a research fellowship from the Charles Stark Draper Laboratory. The author is grateful for the support of the Draper Laboratory and the technical guidance of Peter Kachmar, William Chu, and Timothy Brand of CSDL.

Presentation Overview

- Problem Statements & Objectives
- Space Shuttle Orbital Flight Control System Hardware
- LQG/LTR Autopilot Architecture
- Overview of LQG/LTR Design Methodology
- Simulation Results

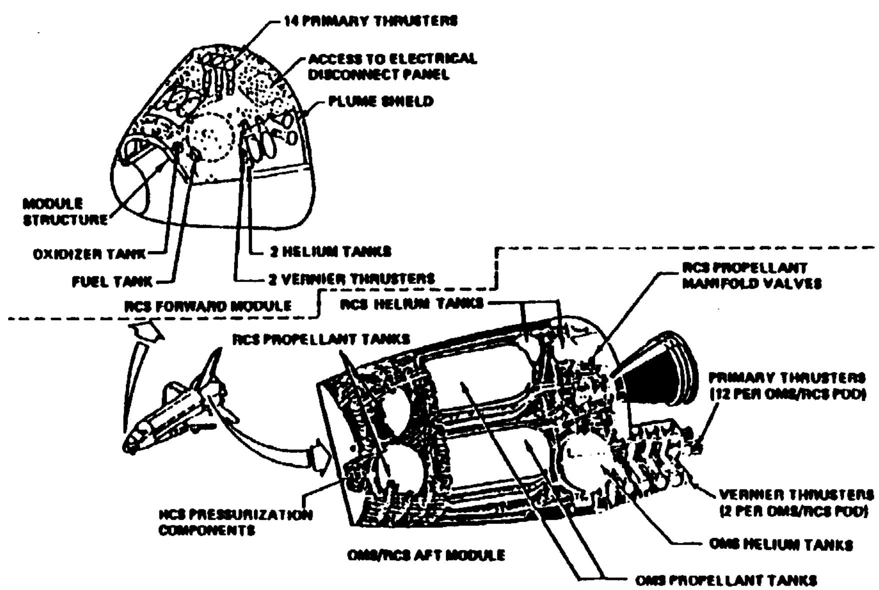
Objectives

- Examine the applicability of the LQG/LTR control system design methodology to the spacecraft proximity operations/docking problem. Capitalize on the beneficial characteristics of LQG/LTR compensators such as decreased sensitivity to sensor noise and robustness to thruster errors and dynamic uncertainty.
- Evaluate the performance of the resulting autopilot for a variety of proximity operations tasks.
- A Space Shuttle Orbiter simulator is used as the technology test bed.

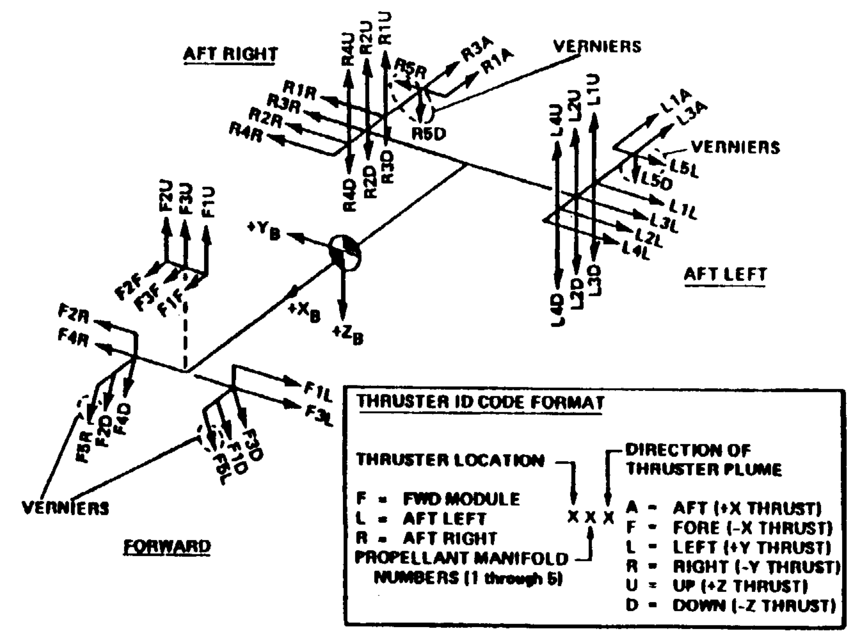
Space Shuttle Reaction Control System

- The reaction control system (RCS) thrusters are located on three pods on the Orbiter.
- The two aft RCS pods each contains an orbital maneuvering system (OMS) engine capable of producing 6,000 lbf of thrust. The OMS engines are used for final orbit insertion, orbit changes, and trim maneuvers. Their large thrust levels preclude their use during proximity operations. The minimum on time of the OMS engines is 2 s.
- The aft RCS pods each also contains 12 primary thrusters. Each primary thruster produces 870 lbf of thrust. In addition, each aft pod contains 2 vernier thrusters which produce 24 lbf of thrust apiece.
- The forward RCS pod contains 14 primary thrusters and 2 vernier thrusters.
- The minimum firing time of the primary and vernier jets is 80 ms.
- The following viewgraph illustrates thruster locations, their plume directions, and identification codes.

Shuttle Reaction Control System Thruster Locations, Plume Directions, and Identification Codes



From JSC-08934 Vol. I, Rev. B



Derived from STS 81-0009

Sensors Used During Proximity Operations

During on-orbit operations, the Orbiter attitude and position states are measured by a set of three inertial measurement units (IMU).

The IMUs are capable of 0-360°/axis attitude range with a 20 arcsec resolution. The IMU inertial attitude is updated by star trackers.

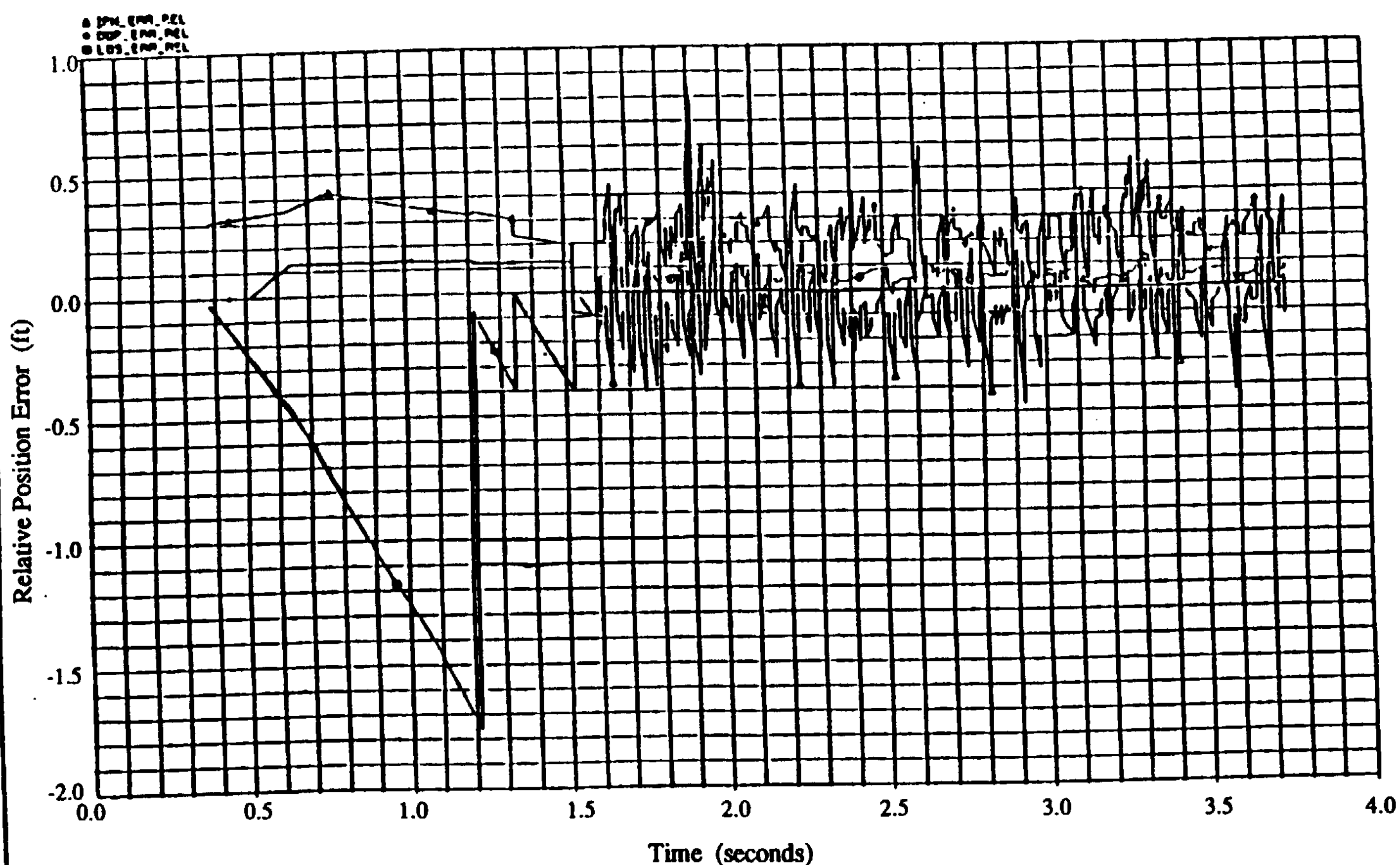
Acceleration can be measured by the IMU within a ± 6 g range with a 1 cm/s^2 resolution.

During proximity operations, the relative position and velocity to the target is critical. Currently, the Orbiter is equipped with a rendezvous radar, which provides the target range, range rate, elevation, and azimuth angles relative to the Orbiter. The rendezvous radar is accurate to about 20 ft and 0.05 ft/s.

For the purposes of this design problem, a more accurate laser rangefinder as assumed. This laser rangefinder was assumed to have a range output noise of 2×10^{-4} ft (1σ), range rate output noise of 2.3×10^{-3} ft/s (1σ), and a angular output noise of 8.6×10^{-5} rad (1σ).

The IMU and laser rangefinder data are processed by the Orbiter navigation system to provide relative Orbiter to target states in the local vertical, local horizontal (LVLH) frame.

Relative Position Measurement Errors Between Orbiter and Target



The Local Vertical, Local Horizontal (LVLH) frame consists of **+x** along the velocity vector, **+z** radially towards the earth, and **+y** in accordance to the right hand rule.

Autopilot Structure

The autopilot architecture can be broken down in several distinct modules. As shown in the following viewgraph, the system consists of a Maneuver Monitor, Maneuver ΔV Logic, Steady State ΔV Logic, LQG/LTR Feedback Compensator, ΔV Loop, and Sensors.

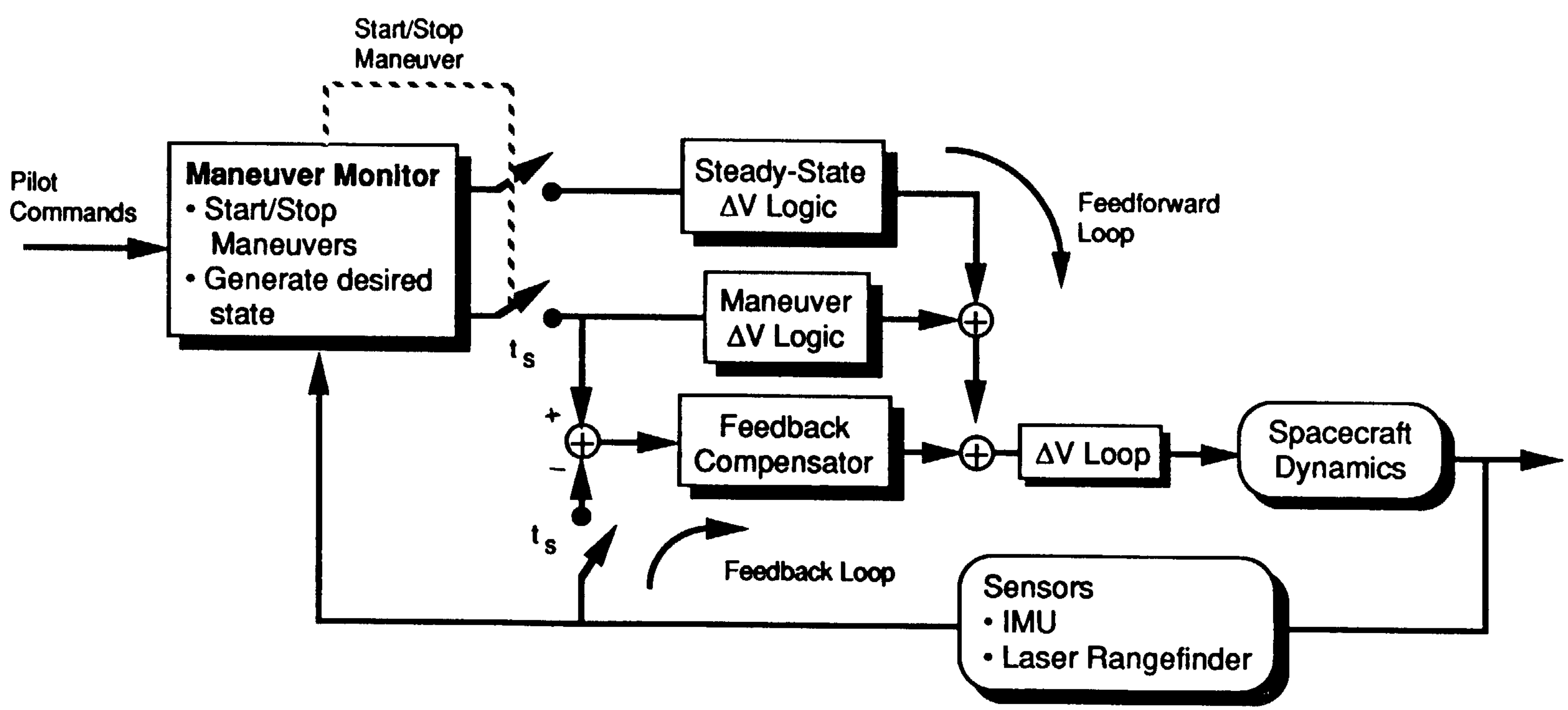
The Maneuver Monitor accepts command inputs from the pilot. These commands may include stationkeeping at the present location relative to the target or performing a translation to a new relative position. The pilot selects a maneuver speed, position and velocity deadbands, and the desired final position. The Maneuver Monitor computes the desired maneuver trajectory and requests a feedforward computation from the Maneuver ΔV Logic which would initiate the maneuver. If the spacecraft trajectory violate the position or velocity deadbands, the Maneuver Monitor would compute a new trajectory and request the Maneuver ΔV Logic to compute a ΔV which would initiate the new trajectory.

Together the Maneuver ΔV Logic and Steady State ΔV Logic make up the feedforward loop of the autopilot. The Maneuver ΔV Logic computes the required velocity change which would initiate or terminate a maneuver. The Steady State ΔV Logic calculate the velocity change required to maintain the spacecraft at non-equilibrium set points (e.g. stationkeeping out of the orbital plane, $\pm y$). Note that in the absence of thruster and sensor error, and dynamic uncertainty, the feedforward loop alone could control the vehicle during coasting arc maneuvers.

The feedback loop consists of dynamic compensators designed using the LQG/LTR methodology. These compensators compare the desired position and velocity from the Maneuver Monitor with the actual Orbiter position and velocity as provided by the sensors. The comparison leads to an error signal which is used to derive a ΔV to bring the spacecraft back to the desired trajectory. The design of the compensators must be robust to sensor noise and thruster errors as well as uncertainty in the vehicle dynamics.

The ΔV Loop accepts velocity change commands from the feedforward and feedback loops in the Local Vertical, Local Horizontal (LVLH) frame and transforms the ΔV commands to the vehicle body frame. A jet select algorithm determines which of the 44 RCS jets should be fired to produce the desired ΔV in the body frame.

Autopilot Structure



Design of the LQG/LTR Compensators

Design of the LQG/LTR compensators is accomplished in three basic steps.

The first step involves the formulation of a target feedback loop (TFL). This TFL should exhibit the desired frequency domain characteristics in the compensated system. The TFL should have high gain at low frequency to maximize performance and roll off at high frequency to decrease sensitivity to sensor noise.

The second step uses the Filter Algebraic Riccati Equation to combine the desired characteristics of the TFL with the natural desirable properties of a Kalman filter loop, such as guaranteed stability.

The final procedure uses loop transfer recovery techniques to formulate a model based compensator (MBC). The MBC substitutes the desirable characteristics of the Kalman filter loop in place of the uncompensated vehicle dynamics.

Design of the LQG/LTR Compensators

1. Formulate a *target feedback loop* (TFL) which exhibits the open loop shape characteristics that are desired for the compensated system.

2. Design a Kalman filter loop which incorporates the frequency domain performance and robustness characteristics of the TFL.

3. Create a *model based compensator* (MBC) to recover the characteristics of the filter loop using loop transfer recovery techniques, thus forming the LQG/LTR compensator.

Detailed discussions of the LQG/LTR design methodology can be found in the following references:

Athans, M., "A Tutorial on the LQG/LTR Method", Proceedings of the American Control Conference, Seattle, Washington, June 1986.

Chen, G., "A Linear Quadratic Gaussian with Loop Transfer Recovery Proximity Operations Autopilot for Spacecraft", Master of Science Thesis, MIT, Cambridge, MA, September 1987.

Selection of the Target Feedback Loop

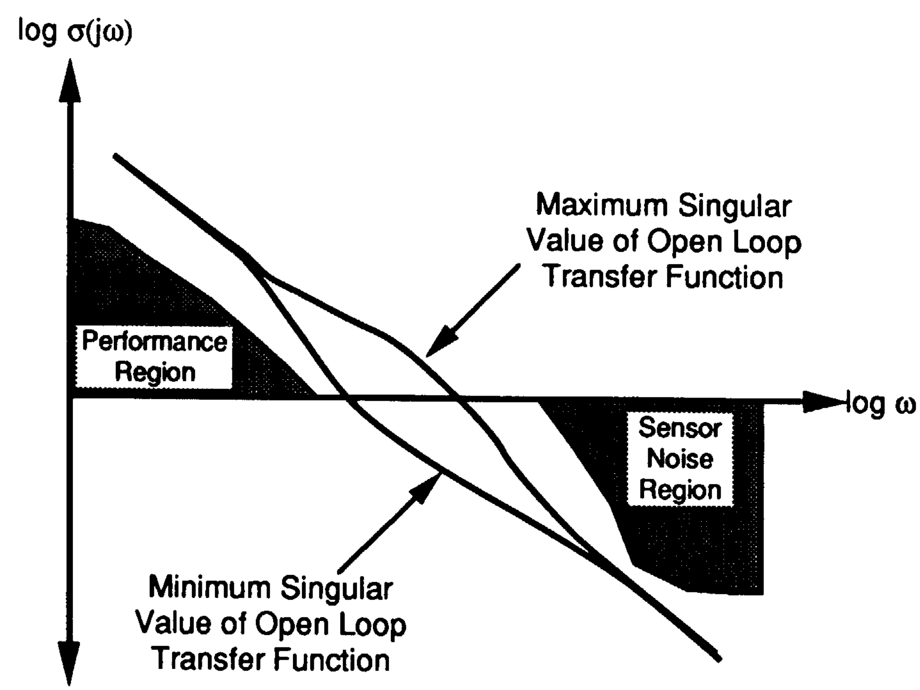
Selection of the TFL is critical since it determines the final system's performance and immunity to sensor noise.

Ideally, the TFL should be increased to maximize feedback gain and provide good performance. This is limited by the desire to attenuate the effects of sensor noise on the system.

Selection of the TFL, as seen in the following viewgraph, can be viewed as a tradeoff between satisfying performance requirements and attenuation of sensor noise.

Notice the -20 db/decade rolloff of the TFL indicating integral action in the forward path. Integrators were deliberately placed in the path to drive errors to zero in the steady state.

Selection of the Target Feedback Loop



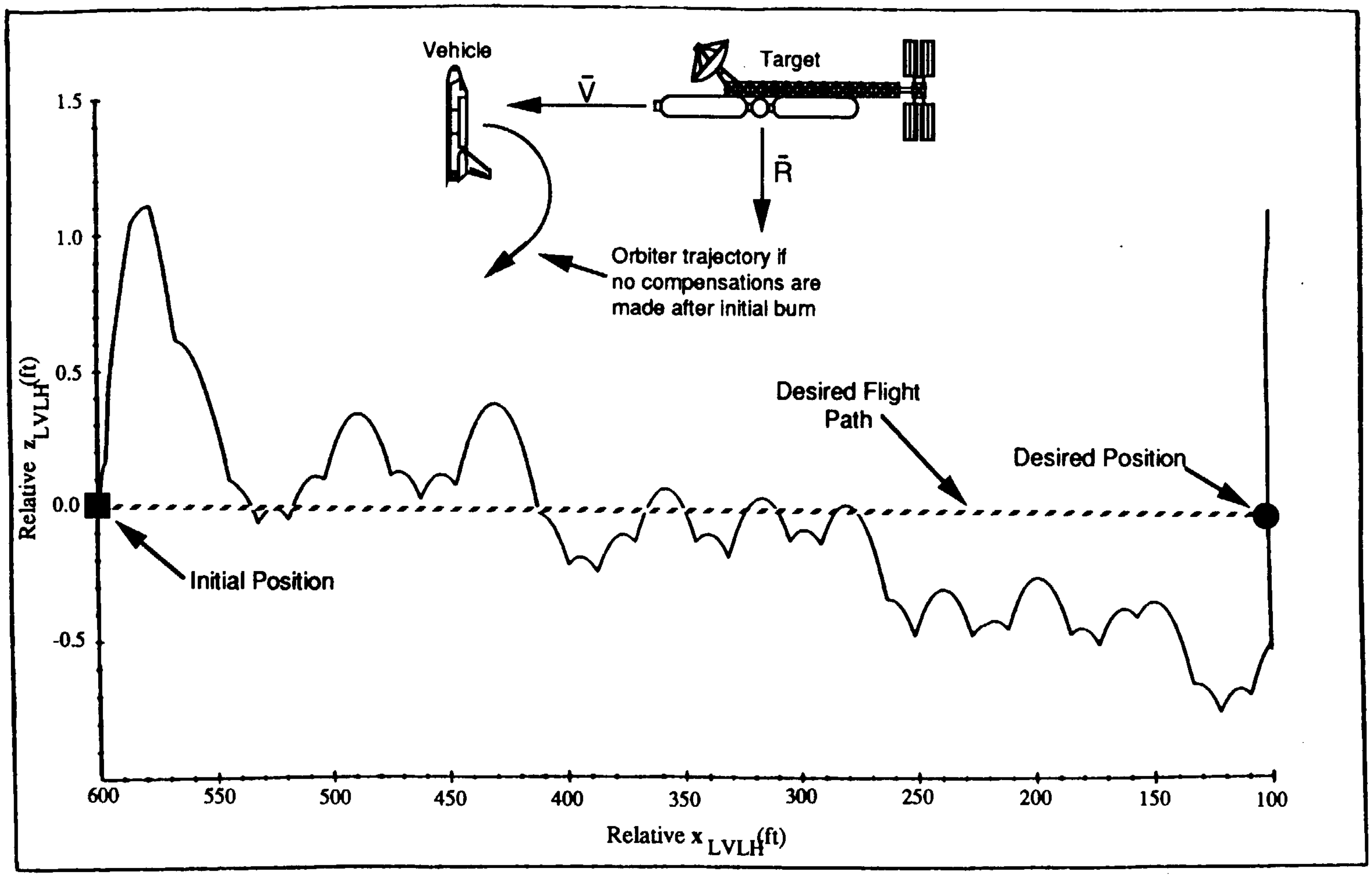
V-Bar Approach

The first simulation begins with the Orbiter 500 ft from the docking port on the Space Station along the velocity vector. The docking port is at position (100, 0, 0) in the LVLH frame. The Orbiter is commanded to translate along the velocity vector (the V-Bar) to the docking port. This type of approach is advantageous since the maneuver can be terminated and equilibrium stationkeeping initiated at any point in the trajectory.

The following plots show the Orbiter trajectory during this maneuver. The first plot shows the Orbiter path in the orbital plane; the second shows the trajectory out of plane. With the exception of the transients at maneuver initiation and termination, the autopilot maintains the Orbiter within 1 ft of the desired trajectory in plane, and 1.7 ft out of plane throughout the maneuver.

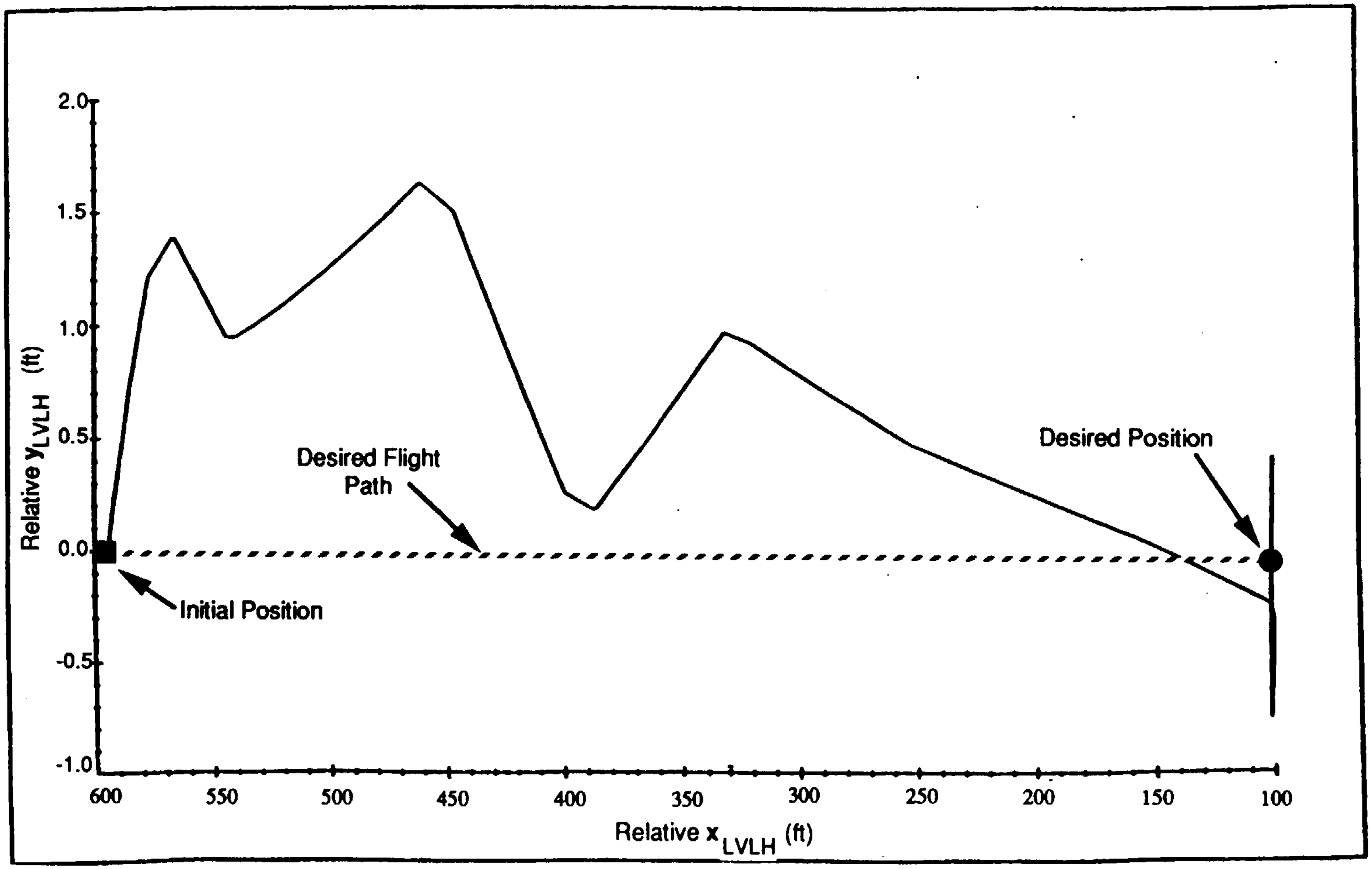
Fuel consumption was 143.08 lbm throughout the maneuver.

V-Bar Approach
In Plane Flight Path



GTC 8/12/1990

V-Bar Approach
Out of Plane Flight Path



PAGE 58 INTENTIONALLY BLANK

GTC 8/12/1990

PRECEDING PAGE BLANK NOT FILMED

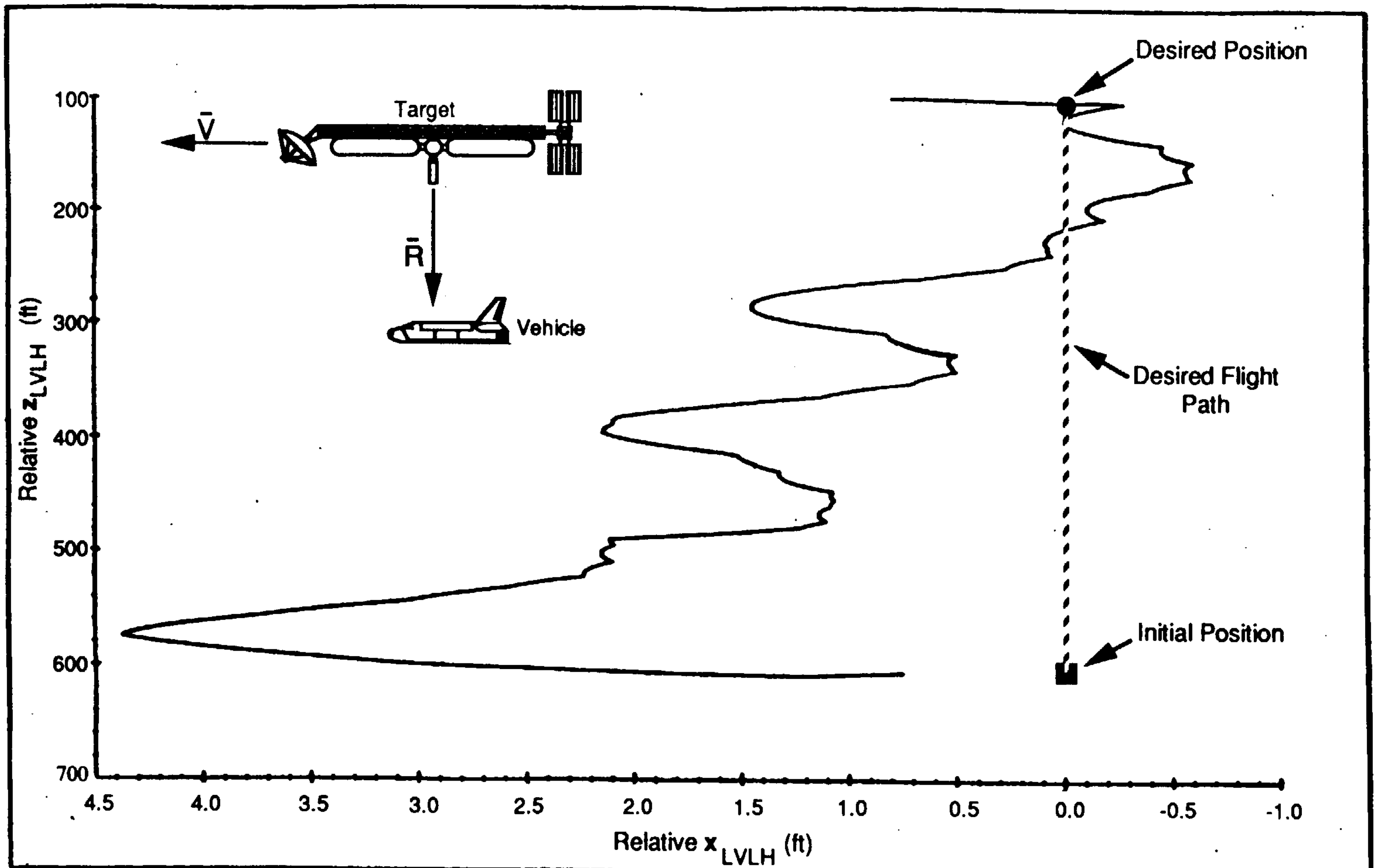
R-Bar Approach

The next simulation has the Orbiter executing a so called R-Bar approach, that is, approaching the target by translating along the radial vector to the earth. This type of approach has the advantage of reducing the thruster plume impingement on the target.

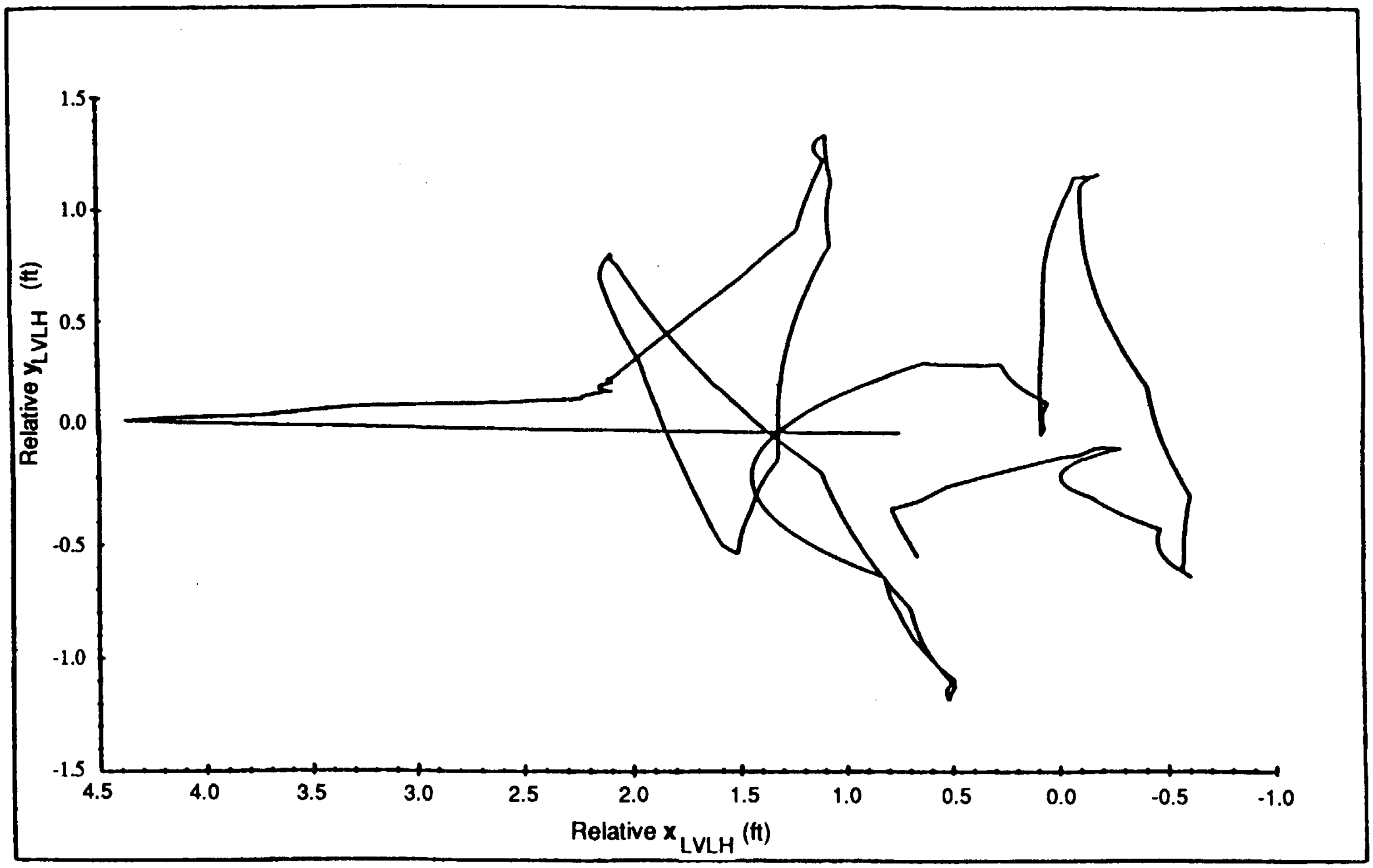
The maneuver begins at position (0, 0, 600) and terminates at (0, 0, 100). In the orbital plane, the Orbiter is maintained within 4.5 ft of the desired path. Out of plane, the Orbiter is maintained within 1.5 ft of the desired path.

Fuel consumption was 212.95 lbm throughout the maneuver.

R-Bar Approach
In Plane Flight Path



R-Bar Approach
Out of Plane Flight Path



PRECEDING PAGE BLANK NOT FILMED

PAGE 62 INTENTIONALLY BLANK

Conclusions

- The LQG/LTR design methodology has validated its applicability to the design of proximity operations autopilots.
- High-fidelity, nonlinear simulations have demonstrated that the LQG/LTR autopilot can provide acceptable performance for a variety of approach and docking scenarios even with state measurements corrupted by realistic levels of sensor noise.

Recommendations for Future Work

- Detailed frequency domain characteristics of the sensor noise should be determined so that the TFL shaping can be done to greater precision, thus maximizing feedback.
- Other design methodologies to formulate the feedback compensators should be explored.

AUTONOMOUS RENDEZVOUS AND DOCKING CONFERENCE
AUGUST 15-16, 1990
HOUSTON, TEXAS

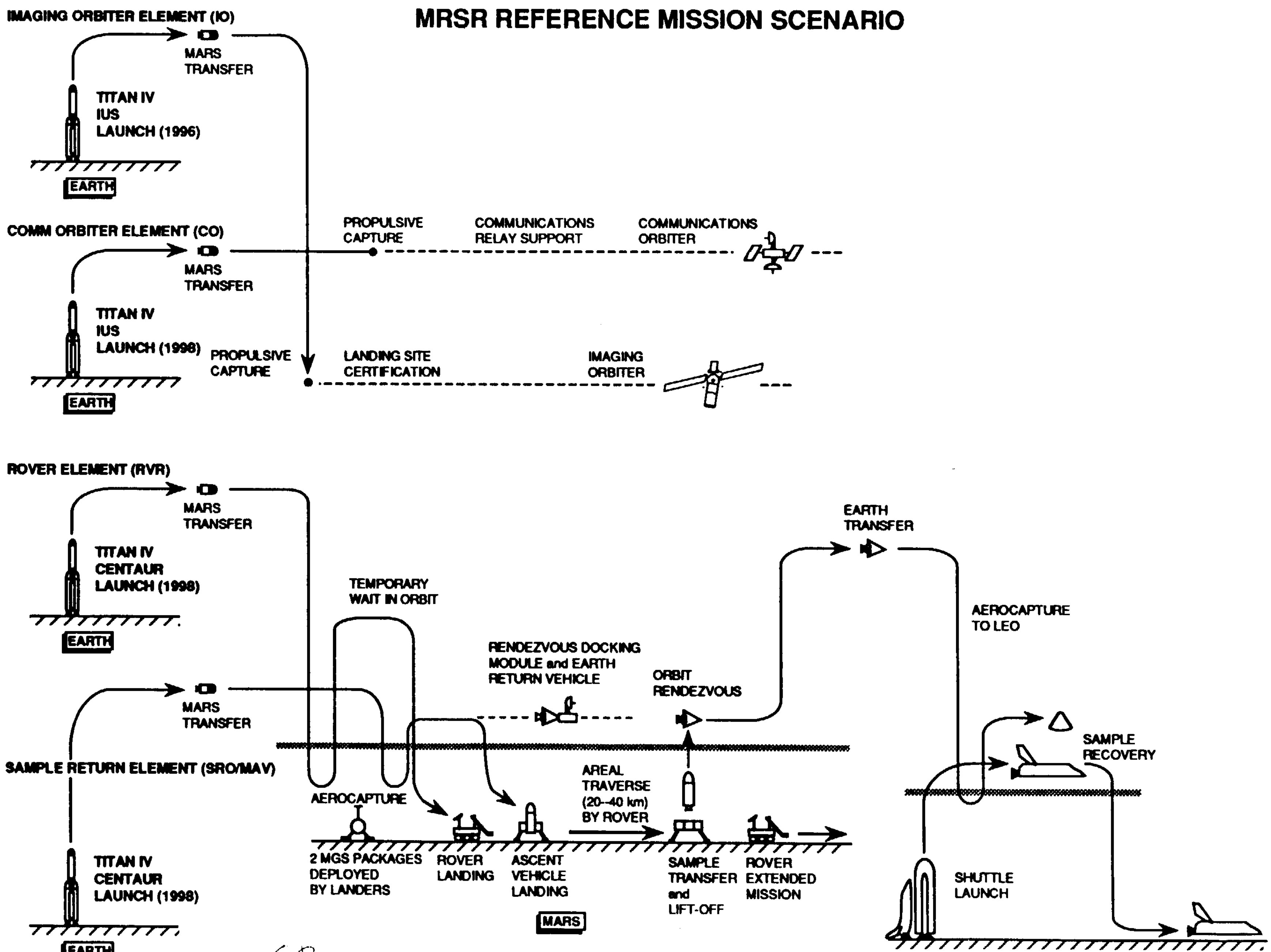
RENDEZVOUS SIMULATION AND ERROR ANALYSIS

NICK G. SMITH
(303)971-6873

MARTIN MARIETTA

NGS REND-1

MRSR REFERENCE MISSION SCENARIO



PAGE 68 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

AUTONOMOUS RENDEZVOUS - OVERVIEW

- MARS ROVER / SAMPLE RETURN (MRSR) SCENARIOS
  - QUICK RENDEZVOUS
  - NOMINAL RENDEZVOUS

- RENDEZVOUS SENSOR COMPARISON
  - SENSOR ACCURACY
  - RENDEZVOUS SIMULATION
  - PERFORMANCE COMPARISON

- MRSR BASELINE
  - NOMINAL RENDEZVOUS SCENARIO
  - COMMUNICATION LINK SENSOR
  - AUTONOMOUS DOCKING

MARTIN MARIETTA

NGS REND-2

QUICK RENDEZVOUS SCENARIO

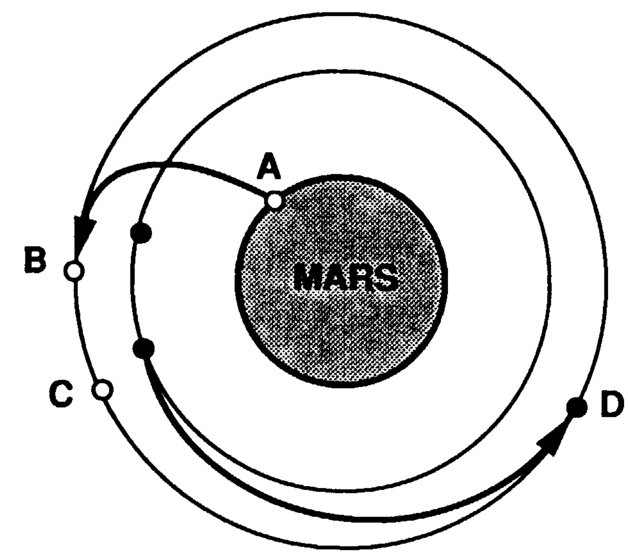
- ● EARTH RETURN VEHICLE (ERV)
- ○ MARS ASCENT VEHICLE (MAV)

A - ERV IN 453 KM CIRCULAR ORBIT, MAV DIRECT ASCENT TO 473 KM

B - MAV ORBIT CIRCULARIZATION, USING ERV-RELATIVE NAVIGATION

C - ERV INITIAL RENDEZVOUS BURN, BELOW AND BEHIND MAV

D - ERV BRAKING MANEUVER, 1 KM IN FRONT OF MAV



MARTIN MARIETTA

NGS REND-3

NOMINAL RENDEZVOUS SCENARIO

A - ERV IN 500 KM CIRCULAR ORBIT, MAV DIRECT ASCENT TO 473 KM

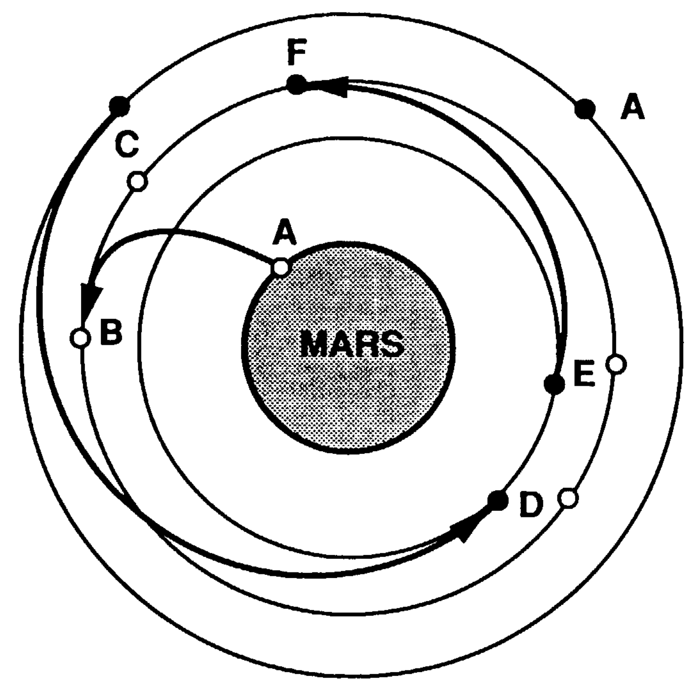
B - MAV ORBIT CIRCULARIZATION, EARTH-BASED TRACKING

C - ERV LOWERED TO 453 KM ORBIT, WHEN NODE ALIGNS WITH MAV

D - ERV ORBIT CIRCULARIZATION, USING MAV-RELATIVE NAVIGATION

E - ERV INITIAL RENDEZVOUS BURN, BELOW AND BEHIND MAV

F - ERV BRAKING MANEUVER, 1 KM IN FRONT OF MAV



- ● EARTH RETURN VEHICLE (ERV)
- ○ MARS ASCENT VEHICLE (MAV)

MARTIN MARIETTA

NGS REND-4

CANDIDATE RENDEZVOUS SENSORS
(1σ ACCURACIES)

	LASER	RADAR	COMM LINK	OPTICAL
RANGE ACCUR.	1 M	0.7% (1 M)	2 M	N/A
RANGE-RATE	0.01 M/S	0.02 M/S	0.04 M/S	N/A
BEARING (DEG)	0.01	0.38	0.25	0.003
MAX RANGE (KM)	62	18	2300	50
MASS (KG)	54.4	40.9	13.9	7.5
POWER (W)	200	62	39	25
VOLUME	0.14 M3	0.079 M3	0.9 M DISH	0.020 M3
HERITAGE	GTE FASTBALL / DELTA SCIENCE	MOTOROLA / OMV	JPL / MAGELLAN	MMC / OFF-THE-SHELF TECH.
COMMENTS	MASS EASILY REDUCED TO 27.2 KG - LT WT STRUCTURE	SKIN TRACK, SIG. MASS & VOL. REDUCT. W/ ADV. TECH.	MRSR BASELINE PRELIM DESIGN, ADDIT. 5.5 KG TRANSPONDER ON MAV (OMNI)	SUNLIGHT BASELINED - INCREASED RANGE MAY ADD LIGHT ON MAV
REFERENCE	BOB JONES / GTE	JOHN LOCKE / MOTOROLA	JOHN FORD / MMC	JOHN MILLER / MMC

MARTIN MARIETTA

NGS REND-5

SENSOR PERFORMANCE COMPARISON ASSUMPTIONS

- RENDEZVOUS INITIAL CONDITIONS
  - MAV IN 473 KM CIRCULAR ORBIT
    - -- POSITION ERRORS (2 KM DOWN, 2 KM RADIAL, 2 KM CROSS / 1σ)
    - -- VELOCITY ERRORS (1 M/S DOWN, 1 M/S RADIAL, 1 M/S CROSS / 1σ)
  - ERV IN 453 KM CIRCULAR ORBIT
    - -- POSITION ERRORS (2 KM DOWN, 2 KM RADIAL, 2 KM CROSS / 1σ)
    - -- VELOCITY ERRORS (1 M/S DOWN, 1 M/S RADIAL, 1 M/S CROSS / 1σ)
  - TRANSFER CENTRAL ANGLE = 130 DEG
  - TRANSFER TIME = 44 MIN

- ERV PROPULSION SYSTEM
  - 4 X 22 N THRUSTERS = 88 N
  - ERV MASS = 937.5 KG
  - ACCELERATION ~ 0.1 M/S2
  - ISP = 311 SEC

- ERROR SOURCES (1σ)
  - IMU INITIAL ACCURACY = 0.07 DEG, DRIFT = 0.03 DEG/HR
  - IMU TO SENSOR ACCURACY = 0.25 DEG
  - ACCELEROMETER BIAS = 17 MICRO G'S, SCALE FACTOR = 70 PPM

MARTIN MARIETTA

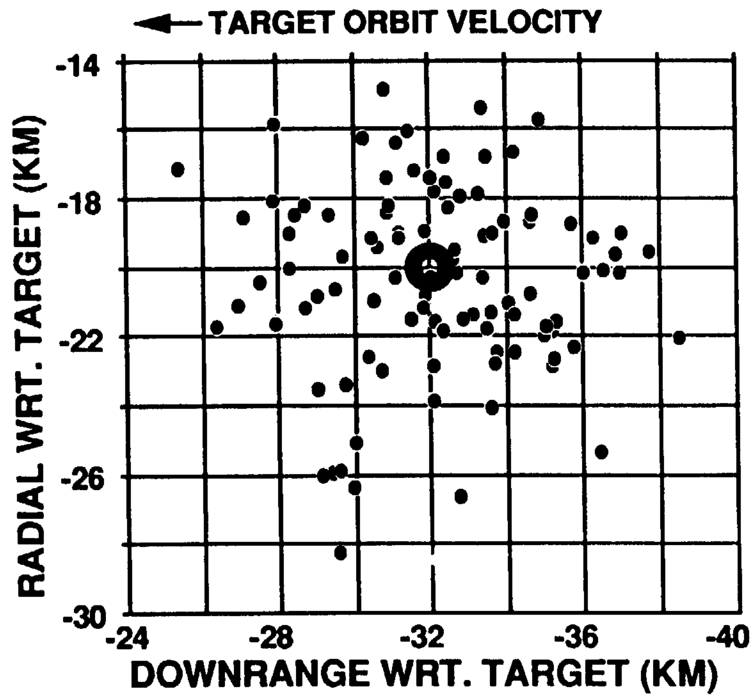
NGS REND-6

RENDEZVOUS SIMULATION

- SIMULATION AND ERROR ANALYSIS OF RENDEZVOUS TRAJECTORIES (SEART)
  - INITIAL STATE ERRORS
  - FINITE BURN MANEUVERING
  - SENSOR ERRORS
  - IMU UPDATE ERRORS AND DRIFT
  - ACCELEROMETER ERRORS

- SENSOR COMPARISON ANALYSIS BASED ON 100 RENDEZVOUS TRAJECTORIES WITH RANDOMLY DISTRIBUTED ERRORS

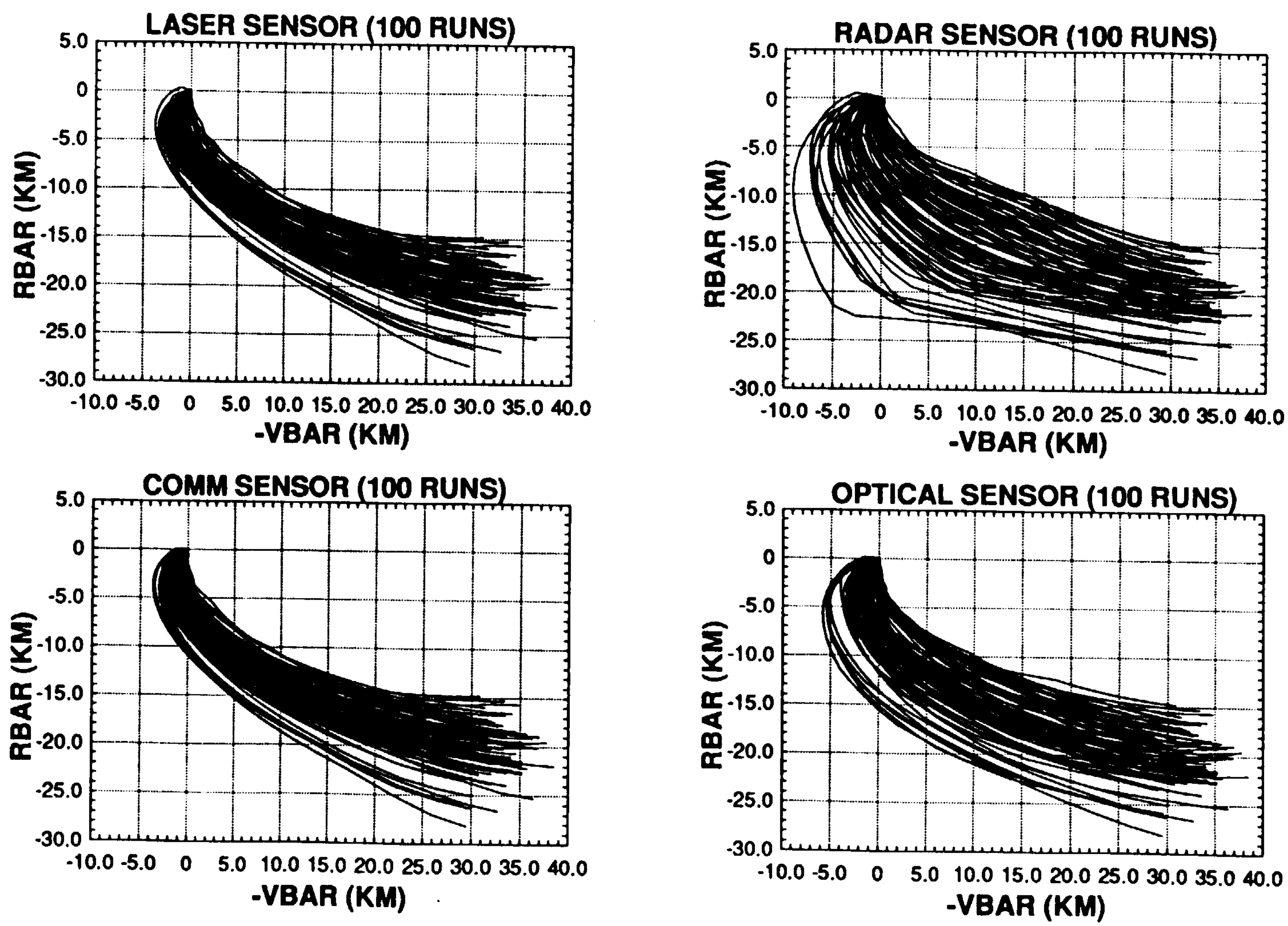
INITIAL TARGET-RELATIVE POSITION
- IN-PLANE COMPONENTS
- RADIAL DISTANCE (RBAR)
- DOWNRANGE DISTANCE (VBAR)
- OUT-OF-PLANE COMPONENT (NOT SHOWN)



MARTIN MARIETTA

NGS REND-7

DISPERSED TRAJECTORIES
(IN-PLANE, TARGET-RELATIVE PROFILES)



MARTIN MARIETTA

NGS REND-8

SENSOR PERFORMANCE COMPARISON RESULTS

PERFORMANCE*	LASER	RADAR	COMM LINK	OPTICAL
DELTA V	34.6 M/S	60.8 M/S	34.1 M/S	38.9 M/S
PROPELLANT	10.6 KG	18.5 KG	10.4 KG	11.9 KG
FINAL POSITION ERROR	78 M	78 M	84 M	181 M
FINAL VELOCITY ERROR	0.21 M/S	0.28 M/S	0.22 M/S	0.25 M/S
MASS	54.4 KG	40.9 KG	13.9 KG	7.5 KG
POWER	200 W	62 W	39 W	25 W
VOLUME	0.14 M3	0.079 M3	0.9 M DISH	0.020 M3

* WORST CASE RESULTS FROM 100-RUN SAMPLE

RENDEZVOUS / DOCKING HAND-OFF CONDITIONS

- POSITION DRIFT ALLOCATION (FOR 1 ORBIT)
  - POSITION DRIFT LIMIT DUE TO POSITION ERROR = 250 M
  - POSITION DRIFT LIMIT DUE TO VELOCITY ERROR = 250 M
  - TOTAL ALLOCATION = 500 M
- POSITION ERROR TOLERANCE
  - RADIAL = 6 M
  - DOWN = 250 M
  - CROSS = 250 M
- VELOCITY ERROR TOLERANCE
  - RADIAL = 0.05 M/S
  - DOWN = 0.01 M/S
  - CROSS = 0.2 M/S
- RECOMMEND INITIATION OF AUTONOMOUS DOCKING IMMEDIATELY AFTER RENDEZVOUS BRAKING

MARTIN MARIETTA

NGS REND-10

CONCLUSIONS

- AUTOMOUS RENDEZVOUS IS FEASIBLE AND OFFERS SIGNIFICANT MISSION MASS SAVINGS

- ACCURACIES DELIVERED BY DEMONSTRATED SENSORS ADEQUATELY SUPPORT AUTONOMOUS RENDEZVOUS

- MARS ROVER / SAMPLE RETURN MISSION BASELINE
  - NOMINAL RENDEZVOUS SCENARIO
    - -- REDUCES ASCENT VEHICLE REQUIREMENTS
    - -- LOWERS MISSION RISK
  - COMMUNICATION LINK SENSOR
    - -- LEVERAGES ALREADY PRESENT SYSTEM
    - -- LOW MASS, LOW POWER
    - -- LONG RANGE
  - INITIATE AUTONOMOUS DOCKING AFTER RENDEZVOUS COMPLETED TO AVOID DRIFT

MARTIN MARIETTA

NGS REND-11

AUTONOMOUS ORBITAL OPERATIONS SOFTWARE TESTBED

presented to:

Autonomous Rendezvous & Docking Conference
NASA Johnson Space Center

presented by:

Daniel C. Bochsler
LinCom Corporation
NASA/JSC Software Technology Branch

August 15, 1990

CONTENTS

- BACKGROUND OF NEED
- AUTONOMOUS ORBITAL OPERATIONS TESTBED REVIEW
- PROGRESS
- TESTBED DESIGN ILLUSTRATIONS
- AUTONOMOUS SYSTEM DESIGN SUMMARY
- DEVELOPMENT PLANS
- RELATIONSHIP TO OTHER PROJECTS
- SUMMARY

Note: Several detailed pages are included in proceedings, but will not be part of presentation.

AUTONOMOUS SYSTEMS DRIVE MANY NEEDS

- IDENTIFICATION OF TECHNOLOGY AND TECHNIQUES TO SUPPORT ON-ORBIT OPERATIONS FOR SPACE VEHICLES AND MANIPULATORS WITHOUT REQUIRING HUMAN INTERVENTION
- THE ABILITY TO INTEGRATE SOFTWARE AND HARDWARE SIMULATIONS TO ACCOMPLISH MISSION DEFINITION AND TO FACILITATE DESIGN OF NEW SYSTEMS

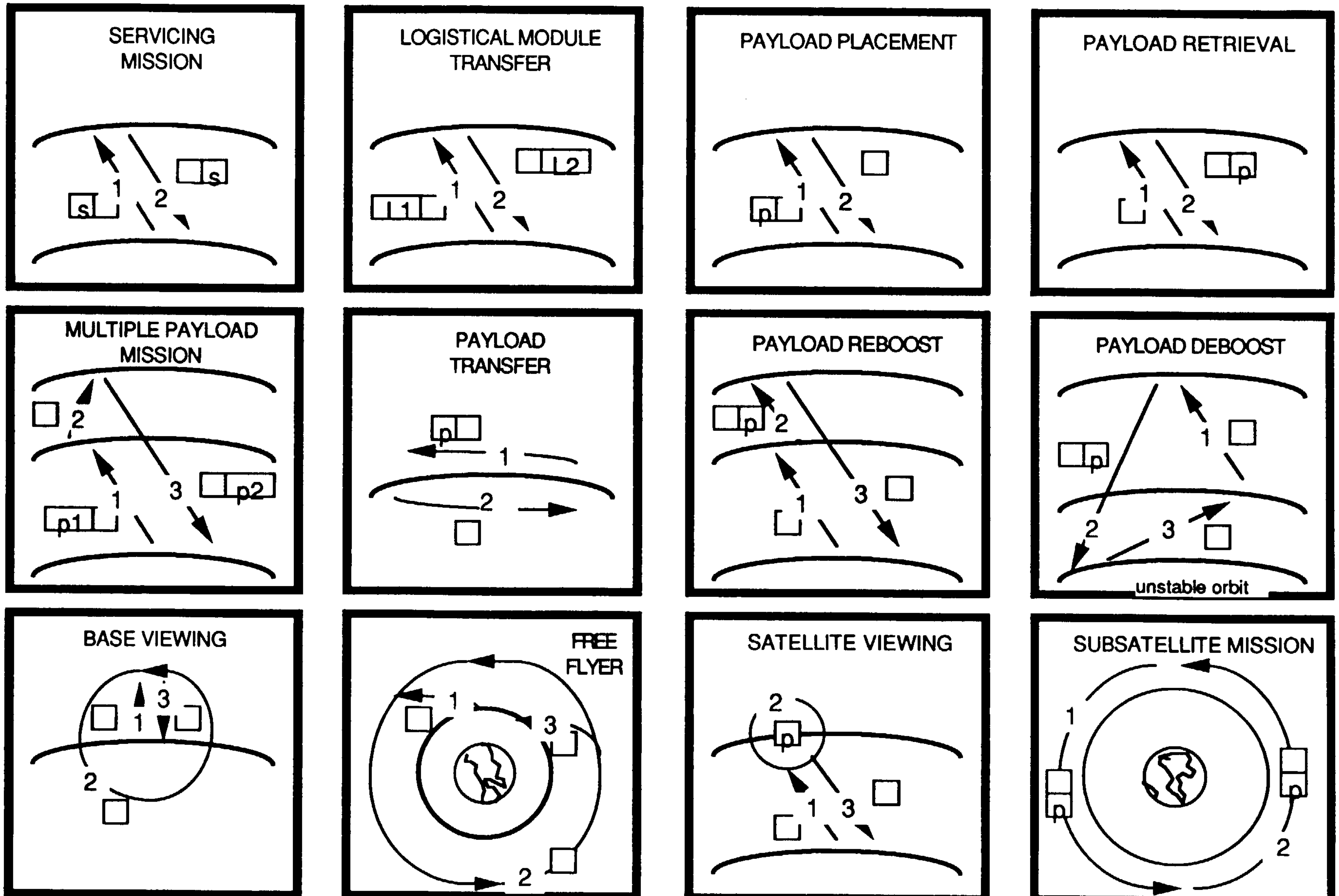
DESIGNERS OF AUTONOMOUS SPACE VEHICLES

- Definition of sensor suites
- Operational control requirements
- Recovery from multiple faults
- Opportunities to reconfigure to accomplish mission without redundancy
- Study operational implications of design features and decisions
- Integrate manipulator and robotic capabilities

MISSION ANALYSTS AND DESIGNERS

- Rapidly investigate operational capabilities of new vehicles
- Examine operational implications of vehicle dynamics
- Create control options to accomplish missions
- Perform "what-if" tradeoffs with varying levels of sophistication
- Verify mission designs from an operational perspective with graphic support

VARIETY OF AUTONOMOUS SYSTEM SCENARIOS



MOTIVATION FOR A SYSTEMS APPROACH

- AN INTEGRATED SYSTEM ENGINEERING APPROACH IS KEY TO SUCCESSFUL PROGRESS THROUGH THE FOLLOWING LEVELS OF TECHNOLOGY APPLICATION:

  LEVEL

  1) BASIC PRINCIPLES OBSERVED AND REPORTED
  2) CONCEPTUAL DESIGN FORMULATED
  3) CONCEPTUAL DESIGN TESTED ANALYTICALLY OR EXPERIMENTALLY
  4) CRITICAL FUNCTION/CHARACTERISTIC DEMONSTRATION
  5) COMPONENT/BREADBOARD TESTED IN RELEVANT ENVIRONMENT
  6) PROTOTYPE/ENGINEERING MODEL TESTED IN RELEVANT ENVIRONMENT
  7) ENGINEERING MODEL TESTED IN SPACE
  8) FULL OPERATIONAL CAPABILITY

- THE WIDE RANGE OF POSSIBLE OPERATIONS NEEDS, CONFIGURATIONS, AND IMPLEMENTATION APPROACHES DICTATES AN INTEGRATED SYSTEMS STRATEGY

- SUCCESS DEPENDS ON PROGRESSING FROM MANUALLY CONTROLLED AND/OR AUTOMATED OPERATIONS TO THE APPROPRIATE DEGREE OF AUTONOMY IN AN EVOLUTIONARY, PLANNED, CONTROLLED, AND WELL-UNDERSTOOD PROGRESSION

OBJECTIVES OF AUTOPS

OVERALL OBJECTIVES

- DEVELOP AN INTEGERATED SOFTWARE TESTBED FOR DESIGN AND ANALYSIS OF AUTONOMOUS SPACE VEHICLES:
  - IDENTIFICATION OF VEHICLE HARDWARE AND SOFTWARE TECHNOLOGY REQUIREMENTS
  - EVALUATION OF HARDWARE/SOFTWARE DESIGNS
  - IDENTIFICATION OF EFFECTIVE COMMAND AND CONTROL APPROACHES
  - PERFORMANCE OF H/W AND S/W TRADE-OFFS BETWEEN OPERATIONAL CONFIGURATIONS
  - INTEGRATION WITHIN CONTEXT OF ORDERLY TRANSITION TO NEW TECHNOLOGIES
  - ASSURING COST EFFECTIVE AND PRACTICAL IMPROVEMENTS

QUESTIONS GUIDING USE OF THE TESTBED

- WHAT IS THE MATURITY LEVEL OF NEW AND EMERGING TECHNOLOGIES WITH RESPECT TO RELIABLE USE FOR VARIOUS KINDS OF SPACE SYSTEMS?

- HOW DO TECHNOLOGIES WHICH OVERLAP EACH OTHER (i.e., MULTIPLE SOLUTIONS FOR A GIVEN PROBLEM) COMPARE AND CONTRAST WITH RESPECT TO SYSTEM/SUBSYSTEM DESIGN AND OPERATIONS PROCEDURES?

- WHAT ARE THE HIGH PAYOFF AREAS FOR USE OF MATURING TECHNOLOGIES?

SPONSORSHIP

- CODE M RTOP MANAGED THROUGH SOFTWARE TECHNOLOGY BRANCH
- NASA STATS CONFERENCE CONSIDERED AUTOPS TESTBED HIGHLY DESIREABLE

BENEFITS OVER PREVIOUS WORK

- GREW OUT OF RECOGNIZED NEED TO FACILITATE AUTONOMOUS RENDEZVOUS & DOCKING MISSION PLANNING AND ANALYSIS (HAD AR&D ORIENTATION FROM BEGINNING)

- TAKES AN INSTITUTIONALIZED APPROACH TOWARD A COMMON ENVIRONMENT
  - REDUCES DIVERGENT/DUPLICATE EFFORTS
  - EASIER TO BECOME AWARE OF PREVIOUS WORK
  - CAN INCORPORATE EXISTING BENCH PROGRAMS

- CONSOLIDATES FEATURES AND CAPABILITIES OF SEVERAL DUPLICATIVE ANALYSIS/ SIMULATION SYSTEMS

- PROVIDES CAPABILITIES TO PERFORM INTEGRATED ANALYSIS OF MULTIPLE TECHNOLOGIES

- A LIBRARY OF VEHICLE, SUBCOMPONENTS, CONTROL SYSTEMS, AND TOOLS TO AVOID DUPLICATED IMPLEMENTATION WORK

- AUTOPS NOT INTENDED AS A PANACEA; RATHER TO REDUCE PROLIFERATION BY PROVIDING A LOT OF WHAT A LOT OF PEOPLE NEED
  - ENABLE OTHER SIMULATION EFFORTS WHICH CANNOT UTILIZE AUTOPS TESTBED TO UNDERSTAND AND UTILIZE APPLICABLE UNDERLYING FEATURES AND IMPLEMENTATION APPROACHES
  - SIMULATION IMPLEMENTATION TECHNOLOGY SOURCE

BASIC TESTBED IMPLEMENTATION REQUIREMENTS

- REAL TIME CONTROL SYSTEMS MUST ACCOUNT FOR CHANGES IN STATE AND ENVIRONMENT DURING "THINK"TIME
- USE INTELLIGENT SOFTWARE AND NETWORK COMPUTING TECHNIQUES TO PARTITION FUNCTIONS AND INCREASE COMPUTING SPEED
- FLEXIBLE ARCHITECTURE TO MATCH/TEST SPACE VEHICLE DESIGNS
- BUILD ON PREVIOUS AND ON-GOING DEVELOPMENTS:
  - NETWORK PROCESSING
  - EXPERT SYSTEM AND FUZZY LOGIC TECHNOLOGY
  - GRAPHICS TECHNOLOGY
  - ROBOTIC/MANIPULATOR TECHNIQUES
  - MISSION PLANNING EXPERTISE
  - OPEATIONS EXPERTISE
  - SIMULATION CAPABILITY
  - OBJECT ORIENTED "PLUG IN" OPERATION
  - STANDARDS FOR PORTABILITY

TESTBED UTILIZATION

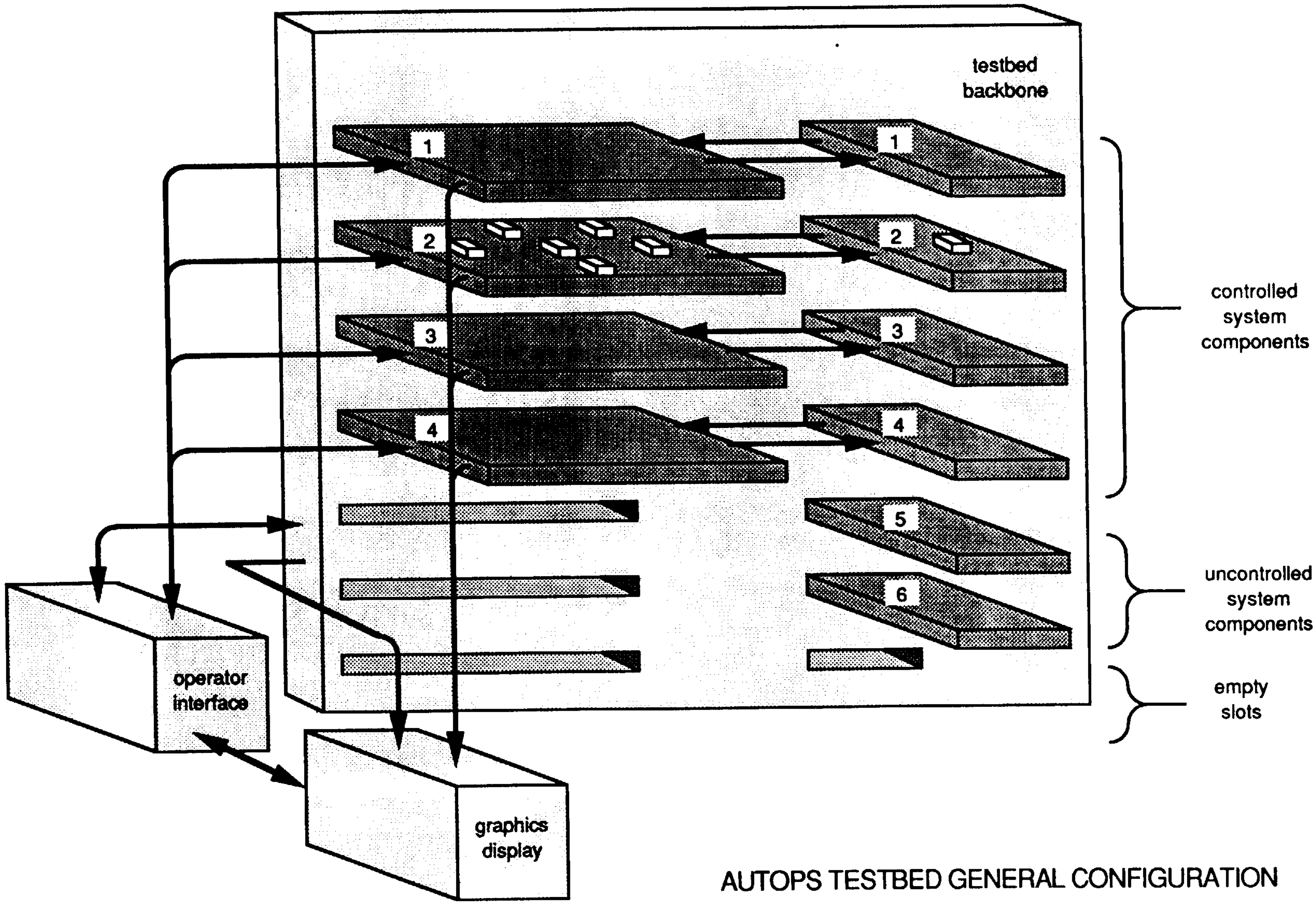
APPROACH

- IDENTIFY THE APPLICABLE FEATURES OF EXISTING TECHNOLOGIES
- FORMULATE A DETAILED CONCEPTUAL MODEL OF THE AUTONOMOUS SYSTEM
- PROTOTYPE THE AUTONOMOUS SYSTEM CONCEPT WITH ANTICIPATED OPERATIONS SCENARIOS (AN EVOLUTIONARY STEP TOWARD A FULL SCALE SOFTWARE SYSTEM)
- DEVELOP AND IMPLEMENT THE LARGER SCALE STRUCTURE OF OPERATIONS ENVIRONMENT
- ADDRESS EVOLVING SPACE SYSTEMS ISSUES:
  - CAPTURE KNOWLEDGE (DESIGN KNOWLEDGE AND OPERATIONS EXPERTISE)
  - IDENTIFY PROBLEM AREAS AND GAPS IN INTEGRATED APPLICATION OF TECHNOLOGIES
  - MAKE MAXIMUM USE OF EXISTING EXPERTISE AND CAPABILITY

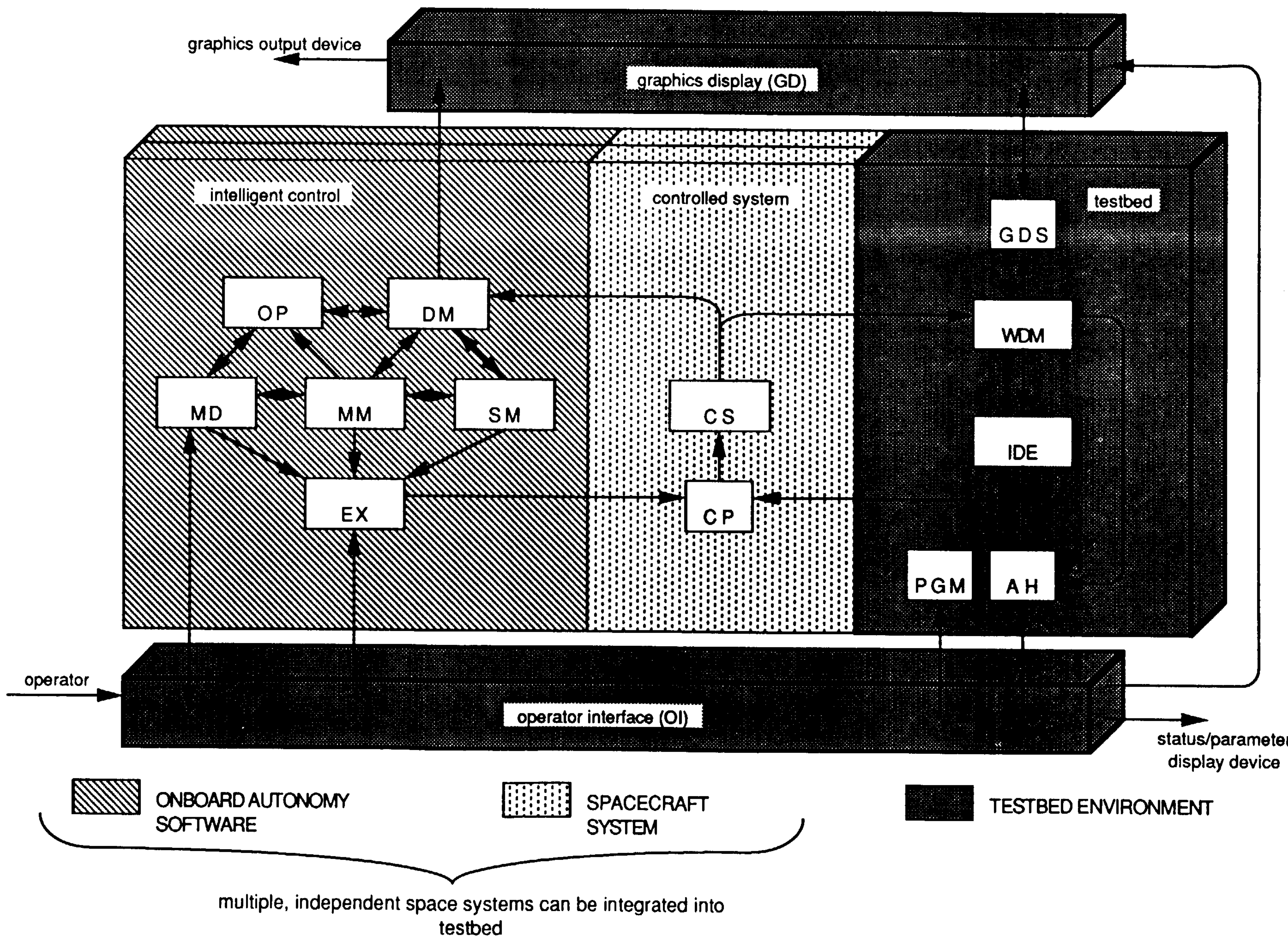
PROGRESS THUS FAR

- BASIC AUTONOMOUS RENDEZVOUS & DOCKING CAPABILITY DEMONSTRATED
- EXPERT SYSTEM PLANNING COMBINED WITH FUZZY LOGIC CONTROLLER FOR PROXOPS
  - TRAJECTORY PLANNING
  - MISSION MONITORING
  - RESPONSE TO ANOMALIES
- COOPERATING EXPERT SYSTEMS DEMONSTRATED:
  - INTERNAL INTELLIGENT CONTROL ELEMENTS
  - ELECTRICAL POWER SUBSYSTEM MONITORING AND FAULT RECOVERY
  - PROPULSION SUBSYSTEM MONITORING
- RAPID INTEGRATION OF TESTBED WITH ANIMATION GRAPHICS
- DETAILED TESTBED AND AUTONOMOUS SYSTEM DESIGN VALIDATED

PLUG-IN MODULES PROVIDE AUTOPS FLEXIBILITY



AUTOPS TESTBED GENERAL CONFIGURATION


graphics output device
graphics display (GD)
intelligent control
controlled system
testbed
OP
DM
MD
MM
SM
EX
CS
CP
GDS
WDM
IDE
PGM
AH
operator
operator interface (OI)
status/parameter display device
ONBOARD AUTONOMY SOFTWARE
SPACECRAFT SYSTEM
TESTBED ENVIRONMENT
multiple, independent space systems can be integrated into testbed

TESTBED COMPONENT ACRONYMS

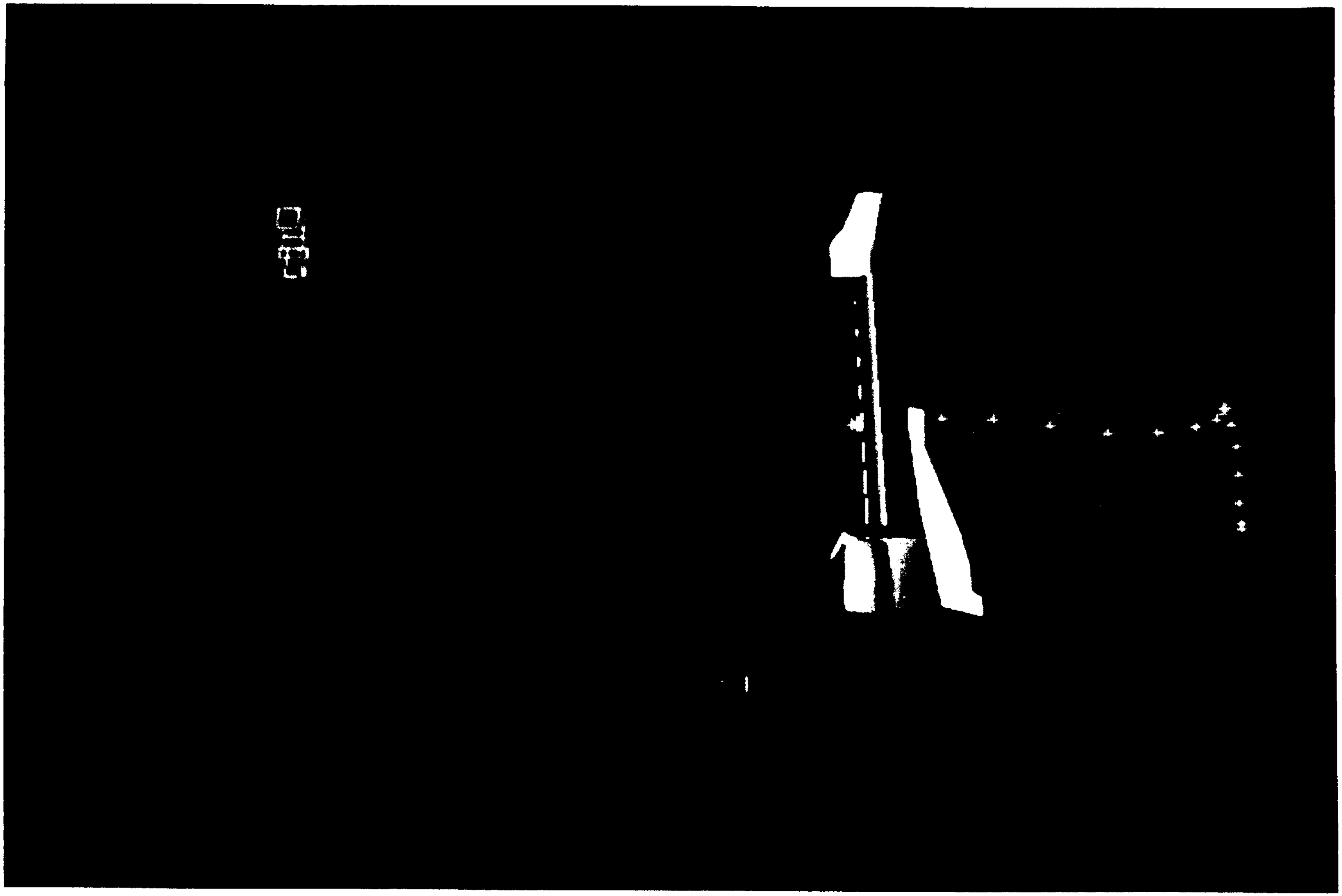
AH...............Anomaly Handler
AUTOPS.....Autonomous Operations System
CP...............Control Preprocessor
CS...............Controlled System
DM...............Data Manager
EX...............Executor
GD...............Graphics Display
GDS...........Graphics Display Supplier
IDE.............Inter-system Data Exchange
MD...............Mission Director
MM.............Mission Monitor
OI................Operator Interface
OP..............Operations Planner
PGM...........Process Generation Manager
SM...............System Monitor
WDM.......... World Data Manager

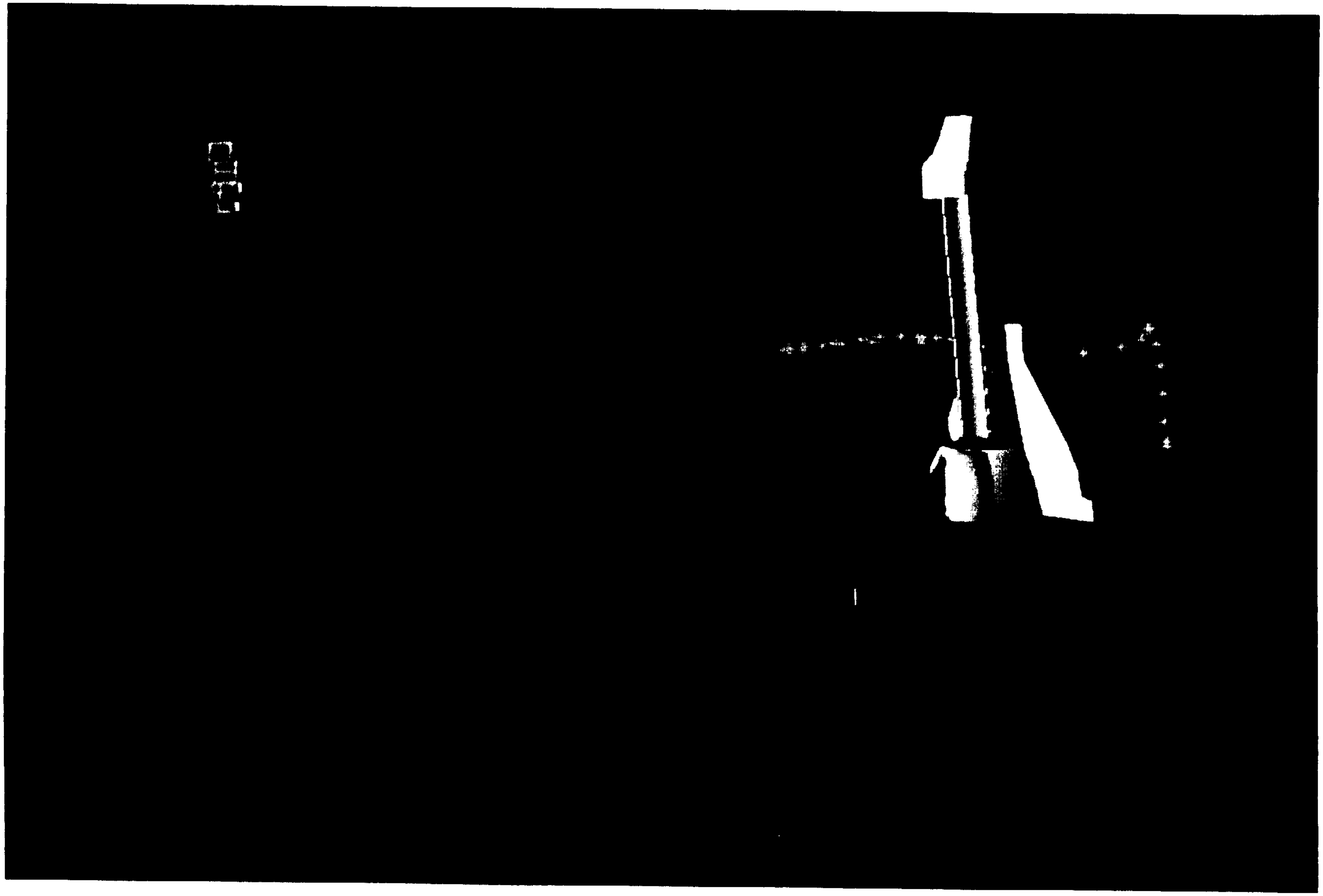
AUTONOMOUS SYSTEM AND TESTBED CONCEPTS SUMMARY (TESTBED COMPONENTS)

COMPONENTS	ROLE IN AUTONOMOUS SYSTEM	GENERAL FUNCTIONS
INTELLIGENT CONTROL	- PRIMARY "INTELLIGENCE" OF THE AUTONONOUS SYSTEM - COLLECTION OF INTERRELATED COMPONENTS CARRYING OUT MISSION - "LEVEL" OF CAPABILITY VARIES WITH: o DEGREE OF OVERALL SYSTEM AUTONOMY DESIRED o COMPLEXITY OF VEHICLE SUBSYSTEMS o LEVEL OF AUTONOMY EMBEDDED WITHIN VEHICLE SYSTEM ITSELF	- MISSION DIRECTOR - OPERATIONS PLANNER - MISSION MONITOR - SYSTEM MONITOR - DATA MANAGER - EXECUTOR
CONTROLLED SYSTEM	- SIMULATION OF THE HARDWARE BEING COMMANDED TO PERFORM A MISSION - INHERENT COMPLEXITY AND CAPABILITY IMPACTS ON INTELLIGENT SYSTEM REQUIREMENTS - PROVIDER OF SENSOR INFORMATION (INCLUDING ERRORS)	- CARRY OUT THE COMMANDS PROVIDED BY THE INTELLIGENT SYSTEM - PROVIDE "TELEMETRY" TO SUPPORT INTELLIGENT CONTROL - DISTORTS SENSORS TO SIMULATE NOISE AND BIAS - SIMULATES ANOMALIES CONSISTENTLY
TESTBED	- INFRASTRUCTURE FOR TESTBED - PROVIDES "UNIVERSE" FOR SIMULATED VEHICLE SYSTEM	- ENSURE SIMULATED VEHICLE SYSTEMS "SEE" THE REAL WORLD APPROPRIATELY - INITIATE USER DEFINED ANOMALIES - SUPPLY "REAL WORLD" GRAPHICS DISPLAY DATA - INITIATE SIMULATION COMPUTATIONS - SYNCHRONIZE PROCESS TIME - ENSURE CONSISTENCY OF ENVIRONMENT AND "REAL WORLD" FACTS - MAINTAIN NETWORK, INITIALIZATION, AND ANOMALY INFORMATION
OPERATOR INTERFACE	- EXTERNAL LINK TO AUTONOMOUS SYSTEM (ONE LINK PER VEHICLE) - INTERFACES FOR VARYING DEGREES OF HUMAN INTERVENTION	- RECEIVES AND PROCESSES INPUTS FROM TESTBED USER - GATHERS & PRESENTS STATUS INFORMATION TO USER
GRAPHICS DISPLAY	- VISUALIZATION OF OPERATIONS ENVIRONMENT - SOURCE OF CUES FOR INJECTING ANOMALIES	- PROVIDE HIGH SPEED ANIMATION GRAPHICS VISUALIZATION - SYNCHRONIZE INFORMATION FROM VEHICLES

ONBOARD AUTONOMOUS SYSTEM DESIGN

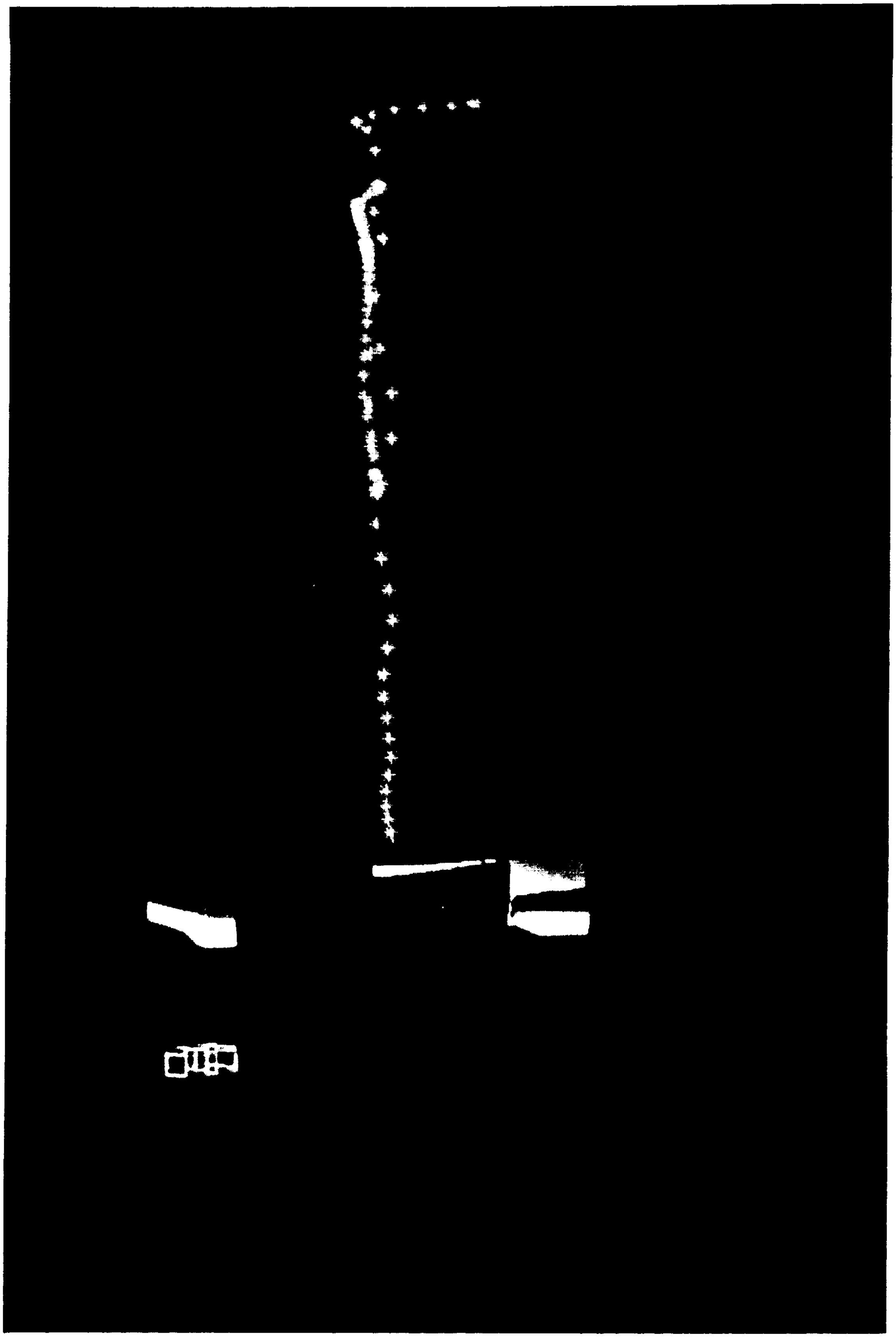
- SELECTED EXAMPLES OF TESTBED GRAPHICS
- 2 MINUTE VIDEO CLIP

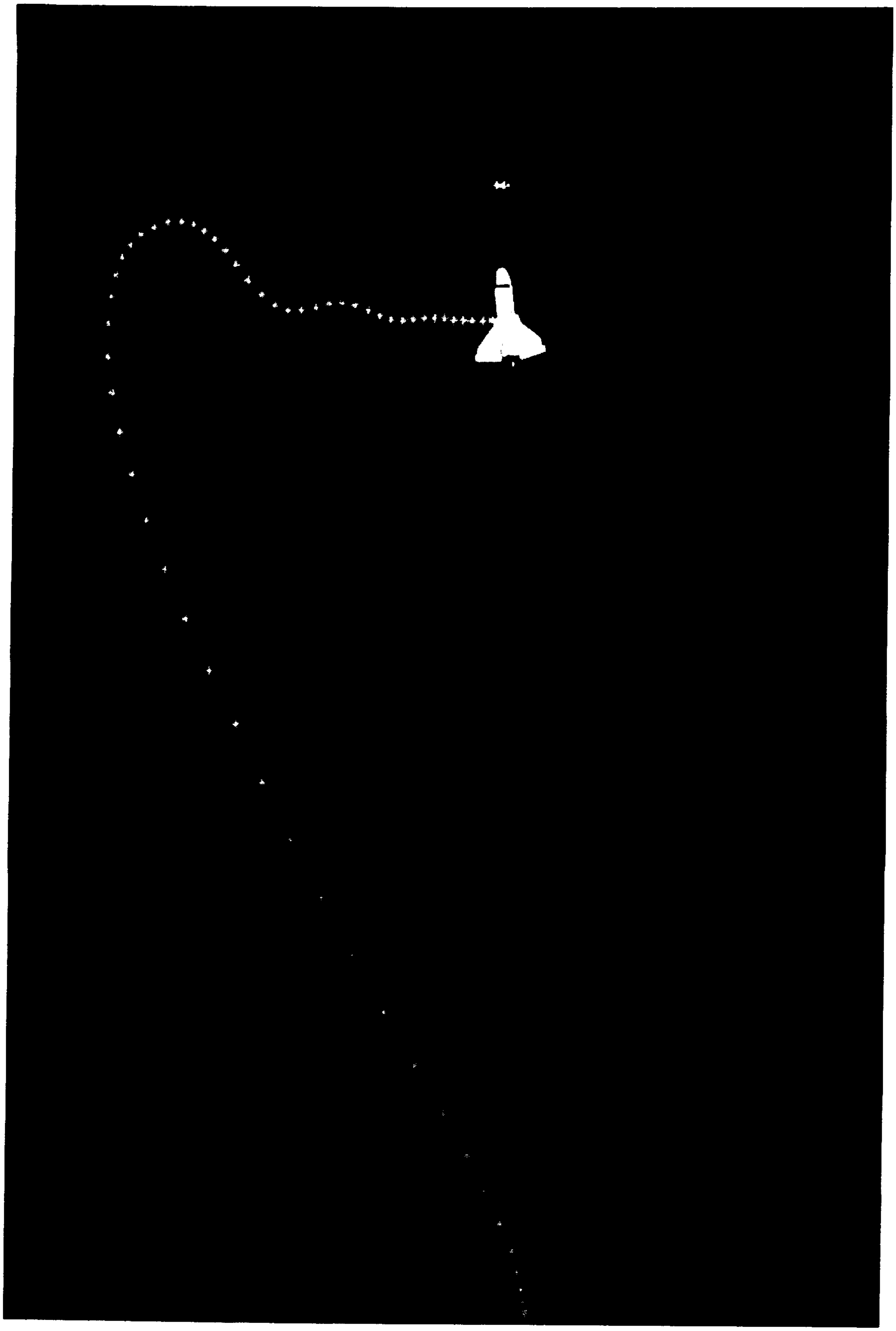




PRECEDING PAGE BLANK NOT FILMED

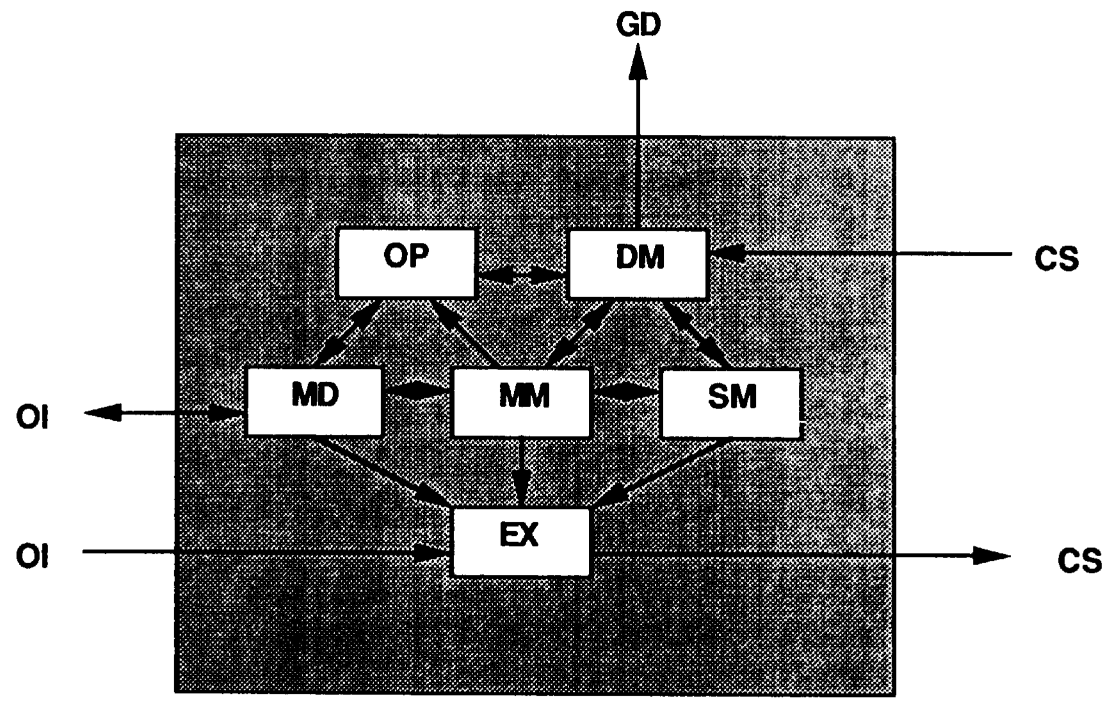
PAGE 102 INTENTIONALLY BLANK





Intelligent Control Design

(top level view)



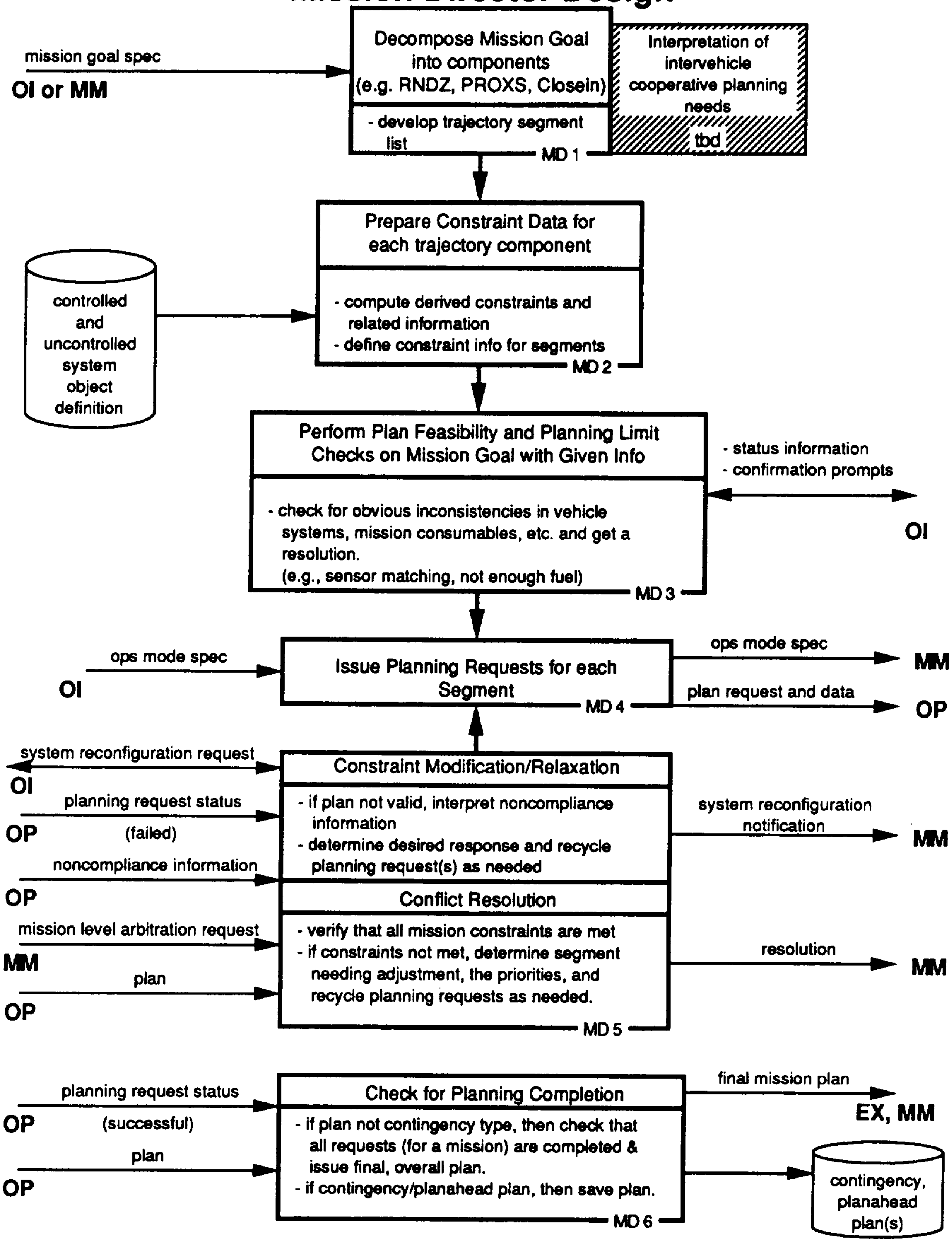
AUTONOMOUS SYSTEM AND TESTBED CONCEPTS SUMMARY (INTELLIGENT CONTROL COMPONENTS)

COMPONENTS	'.E IN AUTONOMOUS SYSTEM	GENERAL FUNCTIONS
MISSION DIRECTOR	- ULTIMATE ARBITER OF CONFLICT IN VEHICLE - SECOND TIER INTERFACE WITH EXTERNAL USER	- UNDERSTAND & INTERPRET OBJECTIVES - PARTITION MISSION GOALS INTO COMPONENT TASKS - DETERMINE OVERALL FEASIBILITY OF MISSION TASKS - ARBITRATE MISSION CONSTRAINT VIOLATIONS AND ACTIVITY CONFLICTS - RECORD CONSTRAINTS IMPOSED BY OPERATOR - INITIATE PLANNING ACTIVITES - POLL OTHER AGENTS FOR MALFUNCTIONS AND "AGENTS' HEARTBEAT"
OPERATIONS PLANNER	- DEVELOPER OF PLANS AND SCHEDULES FOR MISSION TASKS - REPLANNER AS WELL AS PLANNER - CONTINGENCY PLANNER	- VERIFY INFORMATION REQUIRED FOR PLANNING IS AVAILABLE - DEVELOP PLAN TO ACCOMPLISH REQUESTED TASK WITH GIVEN CONSTRAINTS - DEVELOP SAFING PLANS AND CONTINGENCY PLANS
MISSION MONITOR	- MONITOR OF THE PROGRESSION OF MISSION ACTIVITIES - IDENTIFIER OF NEED TO REPLAN - ARBITER OF SUBSYSTEM CONFLICTS NOT IMPACTING MISSION SUCCESS	- COMPARE PLANNED VERSUS ACTUAL MISSION PERFORMANCE - TUNES SYSTEM MONITOR AND DATA MANAGER - DETERMINE APPROPRIATE RESPONSE TO CURRENT SITUATION - HANDLE "KNEE JERK" SAFING ACTIONS - EVALUATE COMPLETION OF MISSION PHASE AND MISSION - AUTHORIZES ALL SUSBSYSTEM CHANGE ACTIONS

AUTONOMOUS SYSTEM AND TESTBED CONCEPTS SUMMARY (INTELLIGENT CONTROL COMPONENTS) (CONT.)

COMPONENTS	ROLE IN AUTONOMOUS SYSTEM	GENERAL FUNCTIONS
DATA MANAGER	- SINGLE RECEIVING POINT FOR ALL DATA FROM CONTROLLED SYSTEM - DISTRIBUTION OF DATA TO OTHER INTELLIGENT COMPONENTS	- RECEIVE DATA FROM CONTROLLED SYSTEM (A VEHICLE) - TRANSFORMS DATA FOR USE BY INTELLIGENT SUBSYSTEMS - MAKE SELECTED DATA AVAILABLE TO GRAPHICS DISPLAY AND OPERATOR - PERFORM SIMPLE ANALYSIS FOR TRENDS, MEANS, NOISE LEVELS, ETC. - STORE DATA FOR RETRIEVAL BY REQUEST
SYSTEM MONITOR	- COLLECTION OF INDIVIDUAL SUBSYSTEM MONITORS PERFORMING HEALTH MAINTENANCE - ARBITER OF CONFLICTS GENERATED BY SUBSYSTEMS IN RECOVERING FROM FAULTS	- LOOK FOR AND INITIATE CORRECTIVE ACTIONS CONTAINED WITHIN A SYSTEM - PROPOSE AND ARBITRATE CHANGES IMPACTING OTHER SUBSYSTEMS - REQUEST AUTHORIZATION FROM MISSION MONITOR FOR CHANGES WITH POSSIBLE MISSION PERFORMANCE IMPACTS - SENDS COMMANDS TO CAUSE SYSTEM CONFIGURATION CHANGES
EXECUTOR	- CENTRAL RECEIVING POINT FOR "COMMAND" INFORMATION - ADMINISTRATIVE "TRANSLATOR" - ONLY COMMAND INTERFACE WITH CONTROLLED SYSTEM	- ESTABLISH SEQUENCE OF CONTROL INFORMATION TO VEHICLE - ADAPT INTELLIGENT CONTROL OUTPUTS TO VEHICLE SIMULATION SPECIFIC FORMATS - EXPEDITE SAFING CONTROL - HANDLE MANUAL OPERATIONS

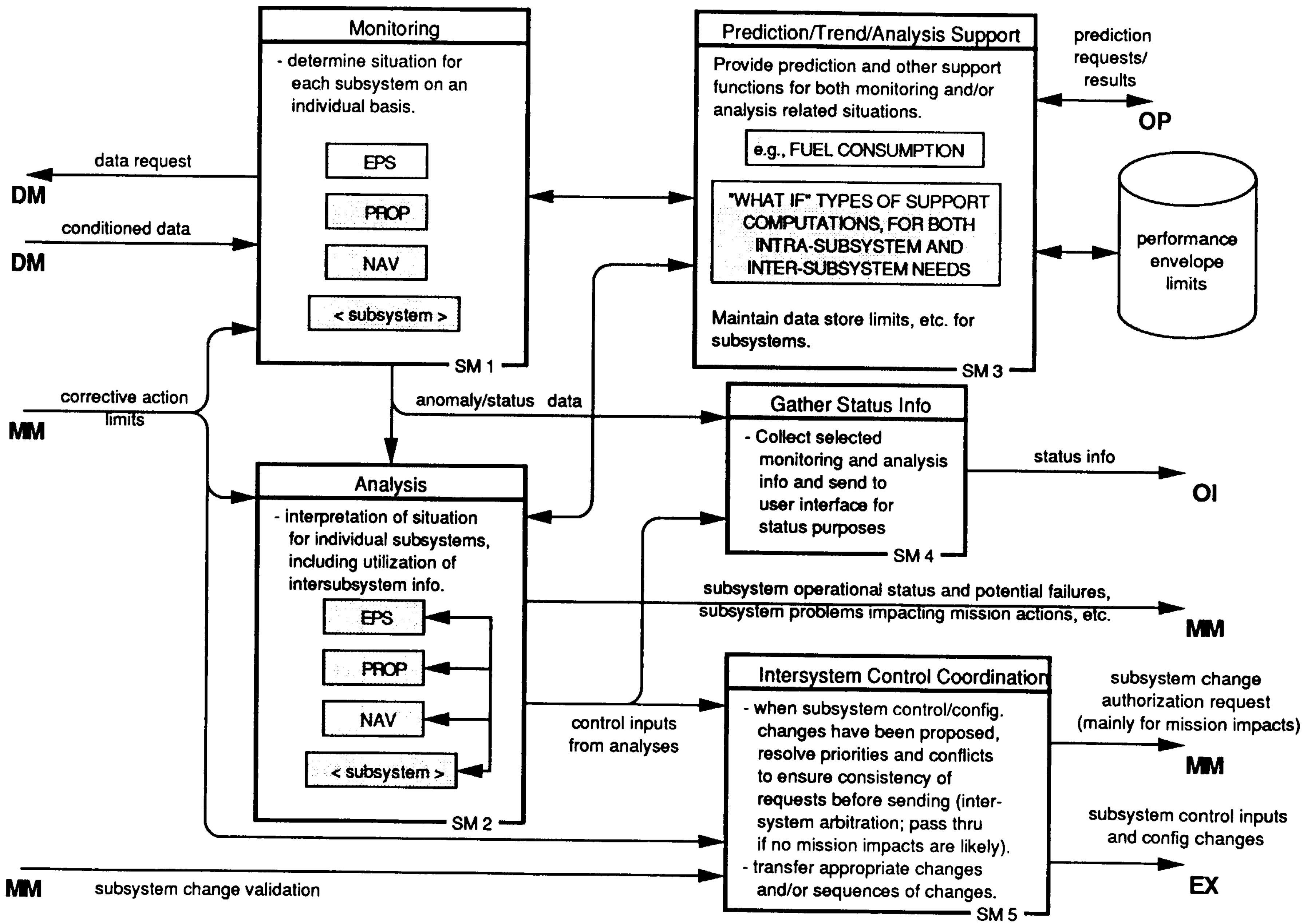
Mission Director Design



II-111

PAGE 110 INTENTIONALLY BLANK

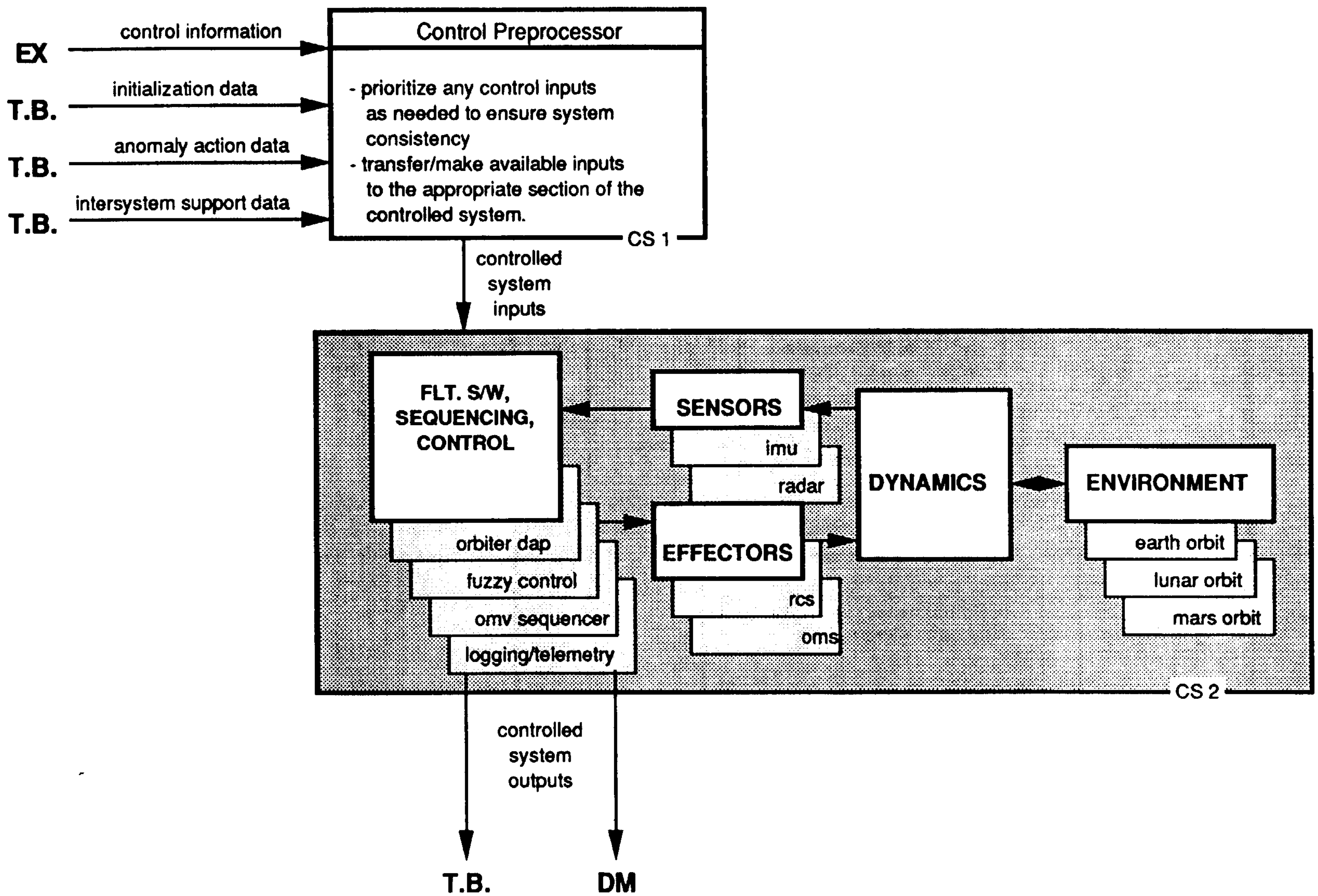
System Monitor Design



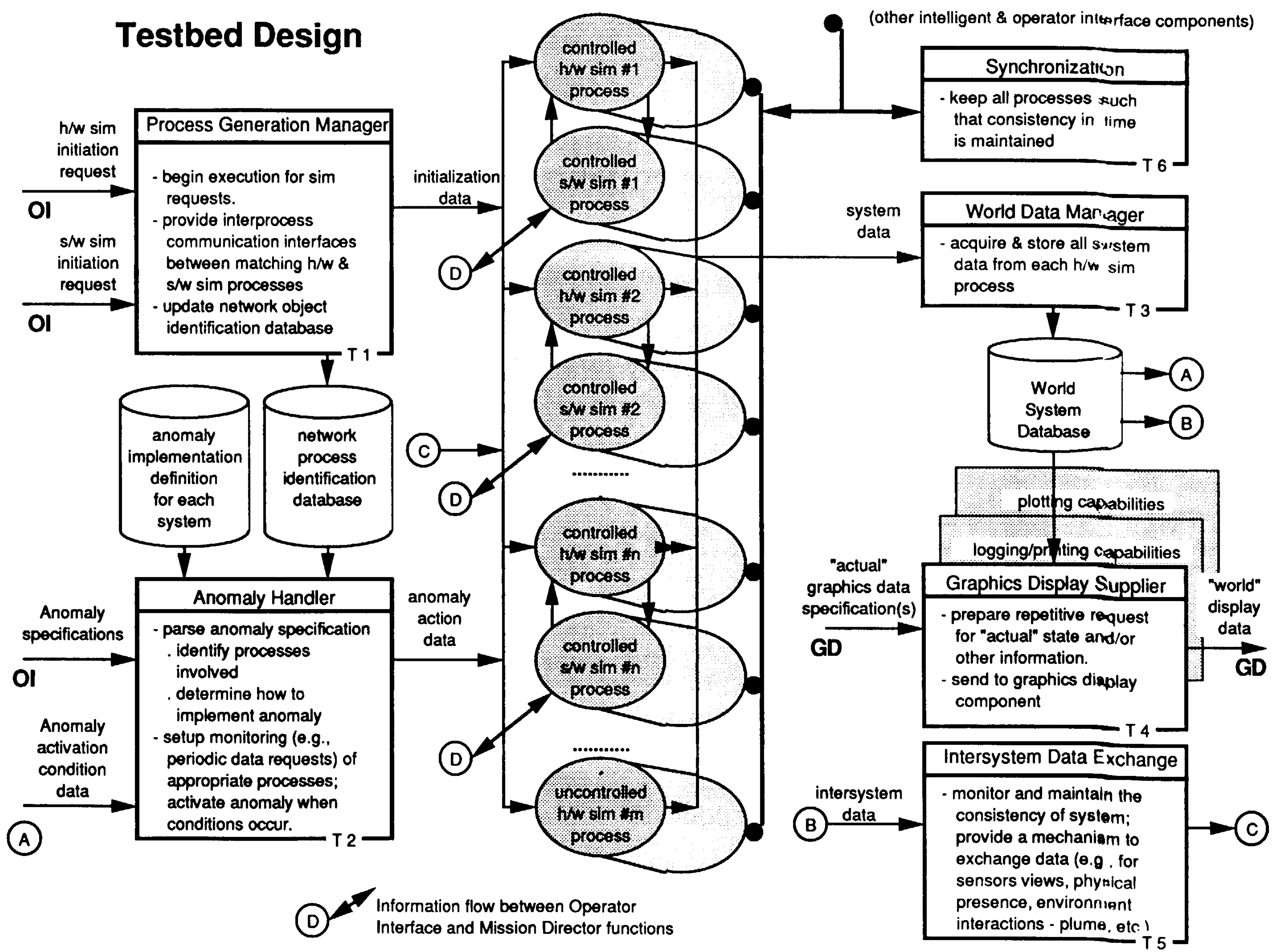
PAGE 112 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

Controlled System Design



Testbed Design



DEVELOPMENT PLANS FOR AUTOPS

- UPCOMING YEAR PLANS IN PROGRESS
  - INTERRUPTION OF CURRENT YEAR WORK (FUNDING)
  - RESUMPTION EXPECTED
  - HOPE TO GATHER ADDITIONAL PERSPECTIVE FROM THIS CONFERENCE

- SYNERGISM WITH OTHER PROJECTS IS BEING EMPHASIZED

- EASE OF USE
  - GENERATE LIBRARIES OF VEHICLES
    - SIMULATION - ORBITAL MECHANICS AND SUBSYSTEMS
    - EXPERT SYSTEMS FOR CONTROL
    - FUZZY LOGIC CONTROLLERS
  - DEVELOP TOOLS TO HELP USERS OF AUTOPS
    - SIMULATIONS OF SUBSYSTEMS
    - EXPERT CONTROL SYSTEMS - COMPILED FOR SPEED
    - USER INTERFACES INCLUDING GRAPHS, INDICATORS, ALARMS, TABLES
    - VEHICLE DEFINITIONS
    - OBJECT AND SIMULATION EDITORS

- FUNCTIONAL USE
  - ACCELERATE AND IMPROVE PRODUCTIVITY OF MISSION AND ENGINEERING EFFORTS
  - CAPTURE KNOWLEDGE OF SPECIFIC AREAS
    - RENDEZVOUS
    - SPACECRAFT DESIGN FOR AUTONOMY

C-2

CONTRIBUTE TO AND DRAW FROM OTHER PROJECTS

- COOPERATING EXPERT SYSTEMS
  - UTILIZE DISTRIBUTED PROCESSING APPROACHES
  - PROVIDE REALISTIC REQUIREMENTS FOR DESIGN TRADEOFFS

- ONBOARD SUBSYSTEMS MANAGEMENT
  - PROVIDE PROVING GROUND FOR DESIGNS AND TESTING OF INTEGRATION ISSUES
  - UTILIZE REQUIREMENTS TO INFLUENCE SUBSYSTEM ARCHITECTURE FEATURES

- AUTOMATION AND ROBOTICS TESTBED DEVELOPMENT
  - UTILIZE ROBOTIC AND MANIPULATION REQUIREMENTS
  - PROVIDE TECHNOLOGY APPROACHES

- JSC PHASE II SBIR INTEGRATED SIMULATION AND GRAPHICS DEVELOPMENT
  - PROVIDE EXAMPLE OF CURRENT AND FUTURE SIMULATION COMPLEXITIES
  - UTILIZE SIMULATION CONTROL INTERFACE DEVELOPMENTS

SUMMARY

- A PROTOTYPE AUTONOMOUS OPERATIONS SOFTWARE TESTBED HAS BEEN BUILT

- USEFUL, LIMITED STUDIES CAN BE CARRIED OUT WITH PROTOTYPE SYSTEM:

  - PERFORMANCE ASSESSMENT OF FUZZY CONTROLLERS FOR ATTITUDE/RATE CONTROL
  - DEVELOPMENT OF REQUIREMENTS FOR INTERFACING COOPERATING ONBOARD SUBSYSTEM MANAGEMENT EXPERT SYSTEMS
  - ASSESSMENT OF ALTERNATIVE PLANNING APPROACHES AND FLIGHT PROCEDURES

- MANY EXISTING APPLICATIONS HAVE BEEN USED (IN WHOLE OR PART) FOR AUTOPS TESTBED DEVELOPMENT - "WHEELS" HAVE NOT BEEN REINVENTED; NEW AREAS ARE BEING PUSHED

- AN AUTONOMOUS SYSTEM DESIGN CONCEPT HAS BEEN FORMULATED AND APPEARS TO HAVE MANY SPINOFFS FOR AUTOMATION IN A BROAD RANGE OF MANNED AND UNMANNED SYSTEMS

SUMMARY (CONT)

- CAPTURE OF DWINDLING RENDEZVOUS EXPERTISE HAS BEGUN
- GRAPHICS REQUIREMENTS FOR AUTOPS ARE PUSHING THE LIMITATIONS OF CURRENT CAPABILITIES
- TECHNOLOGY TRANSFER BETWEEN NASA/AF AND INTRA-NASA ORGANIZATIONS HAS BEGUN
- UPCOMING EFFORTS ARE DIRECTED TOWARD A NETWORKED TESTBED CAPABILITY
- TESTBED WILL FILL AN IMPORTANT ROLE IN TECHNOLOGY DEVELOPMENT AND APPLICATIONS FOR A WIDE SPECTRUM OF UPCOMING AND FUTURE SPACE SYSTEMS

Satellite Servicer System End-to-End Simulation

John A. Cuseo
Space Operations Simulation Laboratory
August 16, 1990

SPACE SYSTEMS

MARTIN MARIETTA

Introduction - Satellite Servicer System End-to-End Simulation

- REAL-TIME INTEGRATED SYSTEMS SIMULATION WITH HARDWARE AND MAN - IN-THE-LOOP PROVIDES HIGH-FIDELITY TESTBED FOR ALL ELEMENTS OF SSSFD PROGRAM
  - autonomous rendezvous and docking
  - supervised autonomous ORU replacement
  - supervised autonomous fluid resupply
  - proximity operations

  Integrated Into Simultaneous End-to-End Simulation

- ALSO INCLUDES SUPPORT ELEMENTS ASSOCIATED WITH THE FLIGHT DEMONSTRATION PROGRAM
  - ground and on-orbit workstation simulators
  - RMS operations for deployment and retrieval of SSS and Target
  - Orbiter prox ops for separation, observation, and rejoin

- SEPARATE MMAG FACILITIES LINKED BY HIGH-SPEED VIDEO/DATA LINKS
  - Space Operations Simulation (SOS) Laboratory
  - Robotics Laboratory
  - Integrated Robotics Facility (IRF) -- planned

- SIMULATION ARCHITECTURE READILY SUPPORTS GROWTH INTO PHASE C/D
  - ground testing and integration of SSS hardware and software
  - operations/procedures development and analysis
  - crew training

MARTIN MARIETTA

Concurrent Engineering Enhanced By Real-Time Simulation

- **REAL-TIME INTEGRATED SYSTEMS END-TO-END SIMULATION IS FOCAL POINT FOR MULTI-FUNCTION TEAMS IN THE INTEGRATION OF SSS DESIGN, OPERATIONS, AND SUPPORT PROCESSES**
- **MAN-IN-THE-LOOP ELEMENT ASSURES CRITICAL HUMAN FACTORS ISSUES (MMI/CREW INTERFACES, PROCEDURES, ETC.) FOR SUPERVISED AUTONOMOUS SYSTEMS ARE ADDRESSED EARLY IN THE PROGRAM LIFE-CYCLE**

SSSFD Program:	**Phase A**	**Phase B**	**Phase C/D**	**Mission Operations**
Simulation Status:	Existing SOS Sim Architecture	End-to-End Simulation Operational		

Increasing Fidelity

- Requirements Development and Analysis
- Concept Design, Development, and Evaluation
- Subsystem Design, Development, and High-Fidelity Testing
- Hardware/Software and Distributed Systems Development and Testing
- Human Factors / Man-Machine (Crew) Interfaces
- Operations/Procedures Development and Analysis
- Mission Timelining
- Crew Training
- Mission Support

MARTIN MARIETTA

SSS End-to-End Simulation Task Elements

SSSFD Task Elements

1. Total SSS System Checkout, Activation, and Shutdown
2. RMS Ops to Deploy and Retrieve Target and SSS
3. Orbiter and SSS Prox Ops for Separation, Observation, and Rejoin
4. Autonomous Rendezvous and Docking
5. Supervised Autonomous Fluid Transfer
6. Supervised Autonomous ORU Exchange

Required Across All SSSFD Task Elements

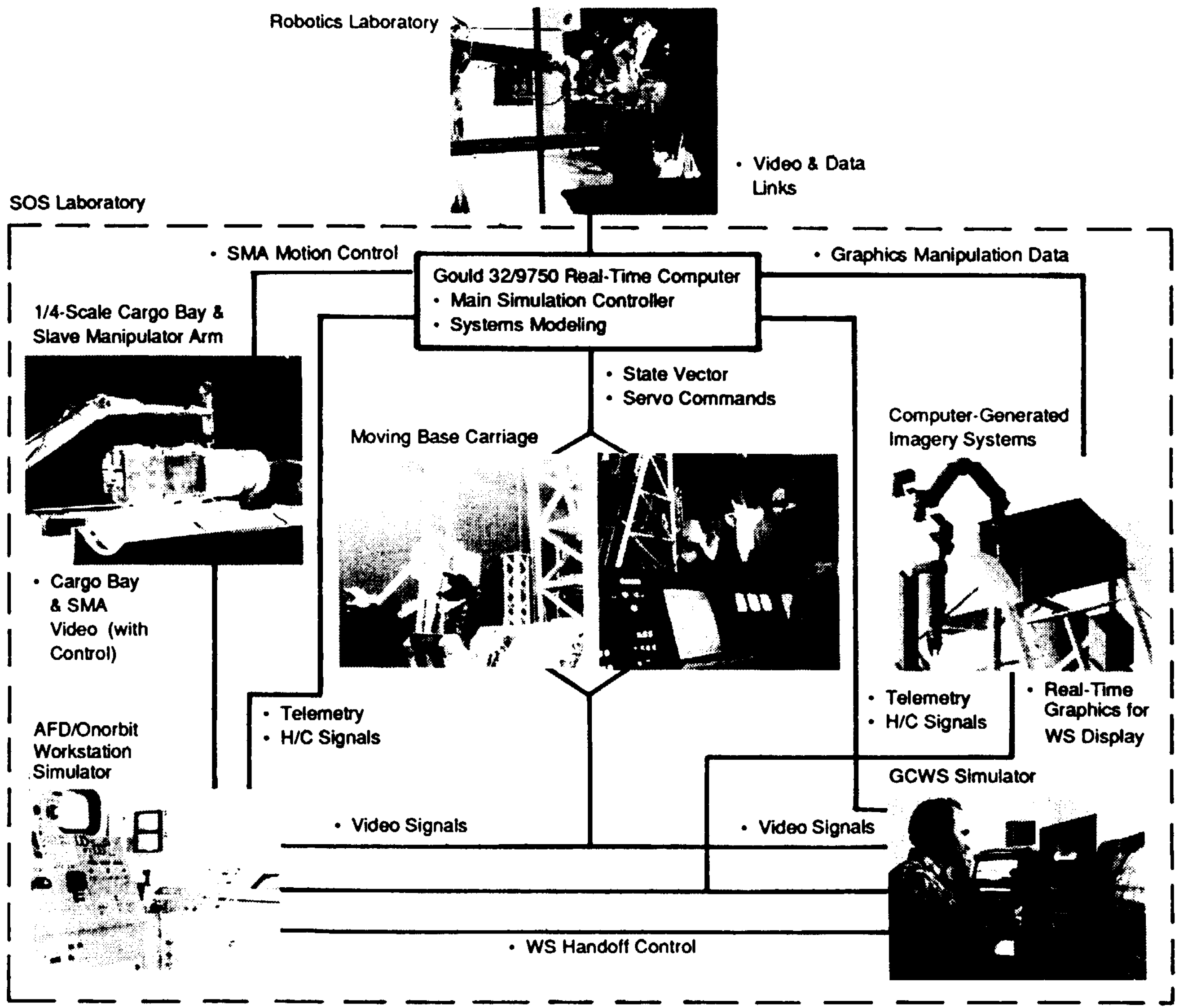
7. Shared Control and Handoff Between Ground and On-Orbit Workstations

Simulation Support Elements

8. SOS Lab Simulation Infrastructure
   - Moving Base Carriage
   - AFD Simulator
   - Ground Control Console
   - Mockups/Models
   - Data Collection S/W
   - Analysis S/W
9. Robotics Lab H/W and S/W
   - Robotic Manipulators
   - ORU Exchange H/W and S/W
   - Fluid Coupling H/W and S/W
   - Task Panels
10. SOS to Robotics Lab Video and Data Interface

MARTIN MARIETTA

SSS End-to-End Simulation Architecture



MARTIN MARIETTA

ARD System Simulation

- VIDEO BASED ARD SYSTEM SIMULATION
  - hardware-in-the-loop (multiple CCD camera configuration)
  - 6 DOF moving base carriage and target gimbals provide relative sensor motion based on high-fidelity spacecraft/environment modelling and dynamics
  - Geometric Arithmetic Parallel Processor (GAPP) for closed-loop image processing
  - inverse perspective algorithm used for position/orientation derivation
  - runs closed-loop and real-time with GN&C

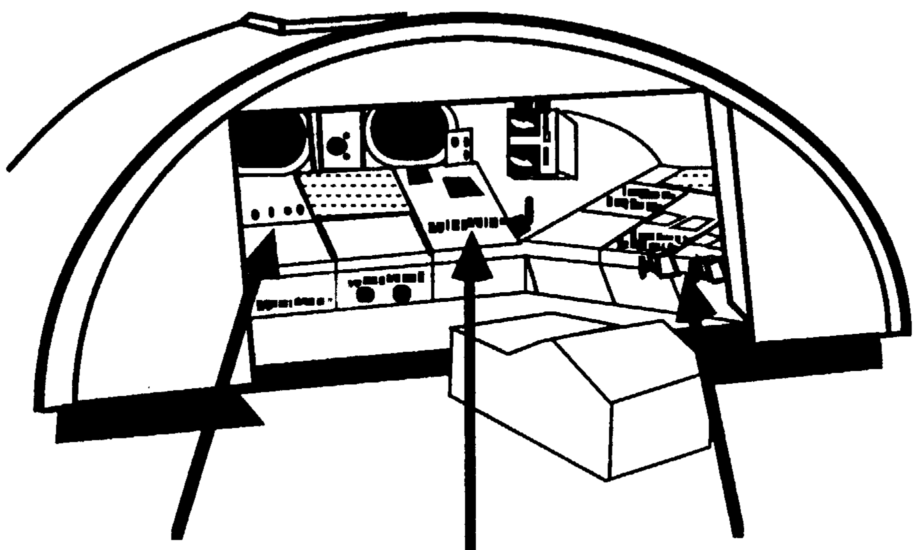
- LASER DOCKING SENSOR (LDS) BASED ARD SIMULATION
  - analytical models of LDS integrated into simulation
  - runs closed-loop and real-time with GN&C
  - moving base carriage provides video sensor motion for supervisory feedback
  - used to assess LDS error characteristics and interaction with control system

* Both systems incorporate dual autonomous/teleoperator control (i.e., supervised autonomous control)

MARTIN MARIETTA

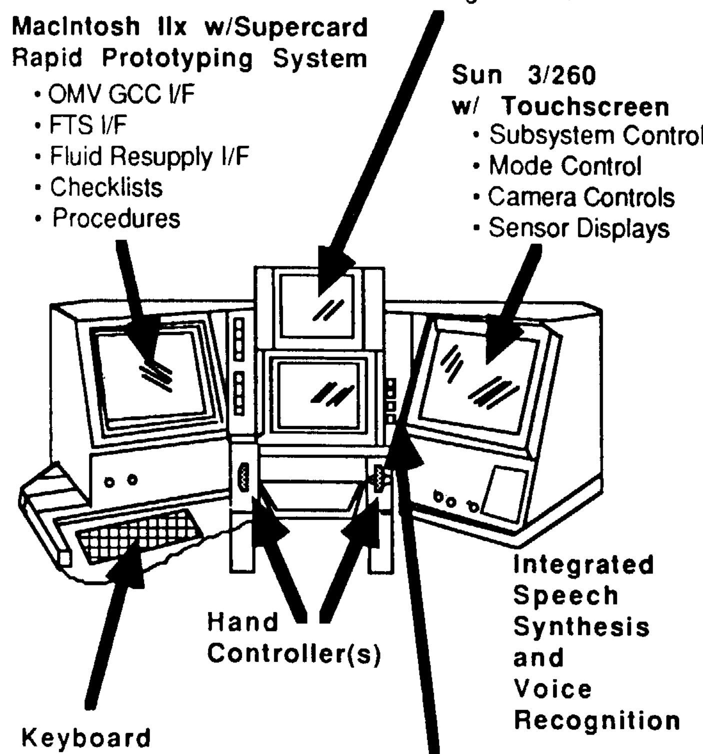
Integrated Workstations for Supervised Autonomous Control

Aft Flight Deck Simulator with SSS On-Orbit Workstation



- Three Integrated AFD Workstations to Meet Key SSSFD Requirement of Simultaneous Operations
- AFD Simulator Integrated with 1/4 Scale Cargo Bay Mockup and Slave Manipulator Arm (1/4 scale RMS)

SSS Ground Control Workstation Simulator



MARTIN MARIETTA

Example of Simulation as a Development Tool

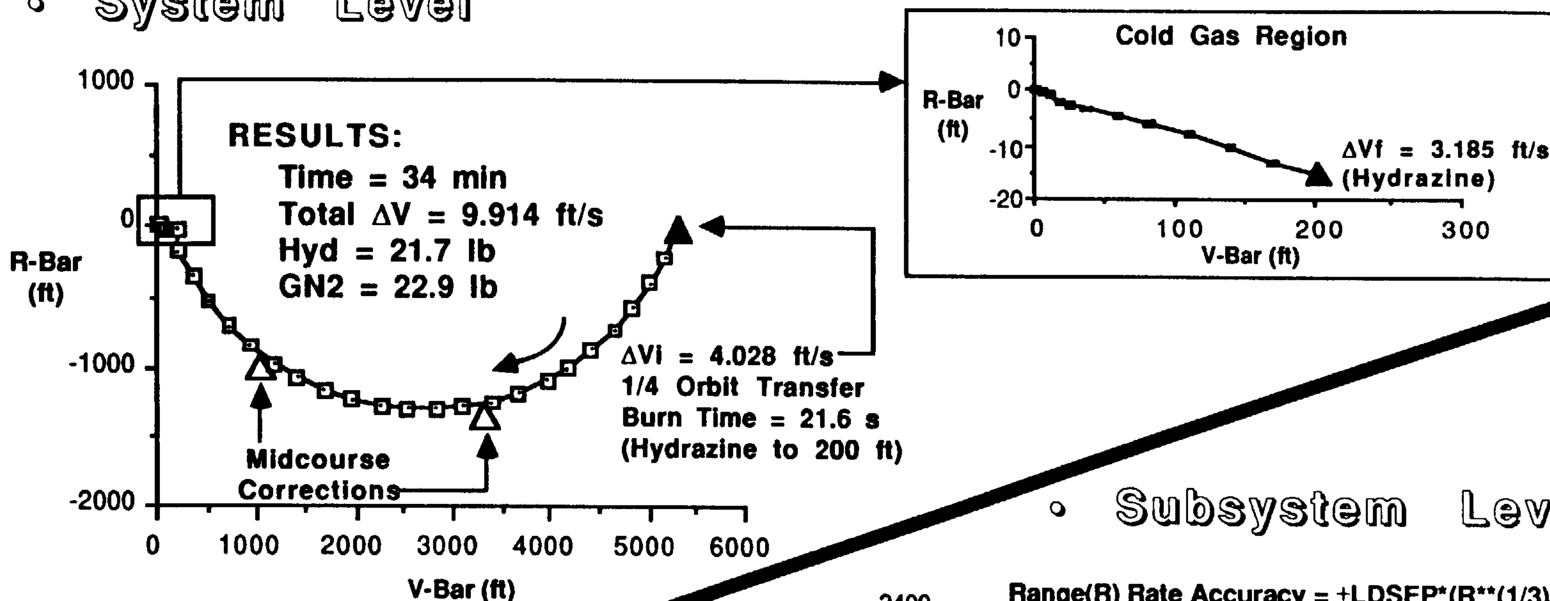
- DEVELOPMENT OF SUPERVISORY AUTONOMOUS/TELEOPERATED CONTROL SYSTEM FACILITATED BY INTEGRATION INTO END-TO-END SIMULATION

Free-Flyer Operations Function Description	Teleoperator Pilot Inputs/Commands	Autonomous System Inputs/Commands	Key Issues
Translational (X,Y,Z) command inputs	Translational Hand Controller Inputs	Translational Autopilot based on LDS derived position/orientation	Rendez/dock profiles; Auto. GN&C algorithms
Rotational (roll,pitch,yaw) command inputs	Rotational Hand Controller Inputs	Auto. Attitude Controller (AAC) based on LDS/IMU derived orient.	Inertial and Target Rel. operational regimes
Attitude (or rate) hold enable/disable	Enable switch; Rot H/C input disables	AAC by IMU until range at which LDS target relative att is available	LDS range for target relative att info; deadbands
Selection of hydrazine (hot) or GN2 (cold) thrusters	Hot/Cold Gas Switch	Selection is a function of range, target motion, and payload	Control authority; Contamination
Selection of continuous or pulse (high or low) thrust	Acceleration Mode Switch	Selection is a function of range, target motion, desired cont. auth.	Control authority; Minimum impulse bit
Selection of direct (nominal) or assist (payload attached) mode	Pilot Mode Switch	Selection is based on grapple sensor indicating attached S/C	Mass properties; Fwd C.G. shift; Controllability
Selection of local or inertial attitude reference	Attitude Reference Switch	LVLH during autonomous rendezvous/docking sequences	Att reference selected to reduce DOFs
Control of grapple mechanism for payload rigidization	Grapple Controls	Control based on contact sensors in grapple mechanism	Deactivation of TV att control after docking

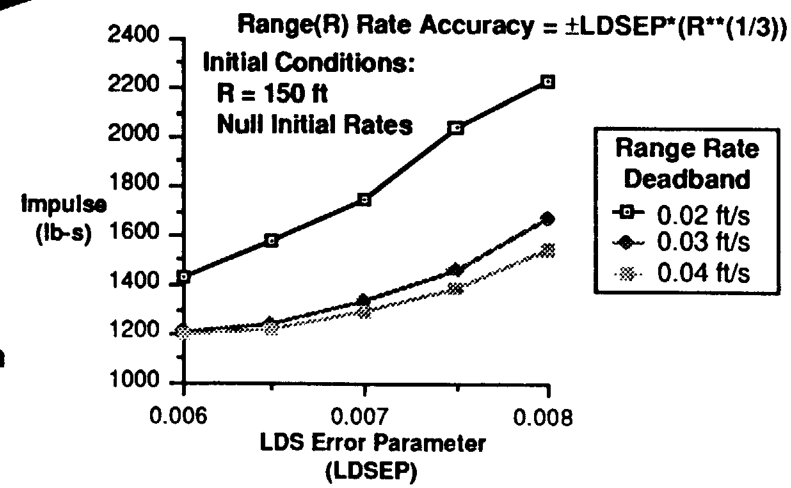
MARTIN MARIETTA

Example of Simulation as an Analysis Tool

- System Level



- Subsystem Level



- Laser Sensor Models Integrated Into Sim
- LDS Error Characteristics And Interaction With Control System Are Key Issues
- Figure Depicts Two-Factor Interaction Between LDS Errors and Deadbands

MARTIN MARIETTA

Summary - Satellite Servicer System End-to-End Simulation

- End-to-End Simulation Operational for Phase B Start
- High-Fidelity Ground Based Testbed for All SSS Tasks
- H/W and Man-in-the-Loop for Supervised Autonomy
- Real-Time Simulation Development Typical of Flight System
- Low Cost Growth Potential to Phase C/D and Ops Support

MARTIN MARIETTA

Hermes and Columbus Rendezvous Control System

C. M. Pauvert; J.M. Pairot
MATRA ESPACE
Toulouse
France

Dr. W. Fehse; A. Tobias
EUROPEAN SPACE AGENCY
Noordwijk
The Netherlands

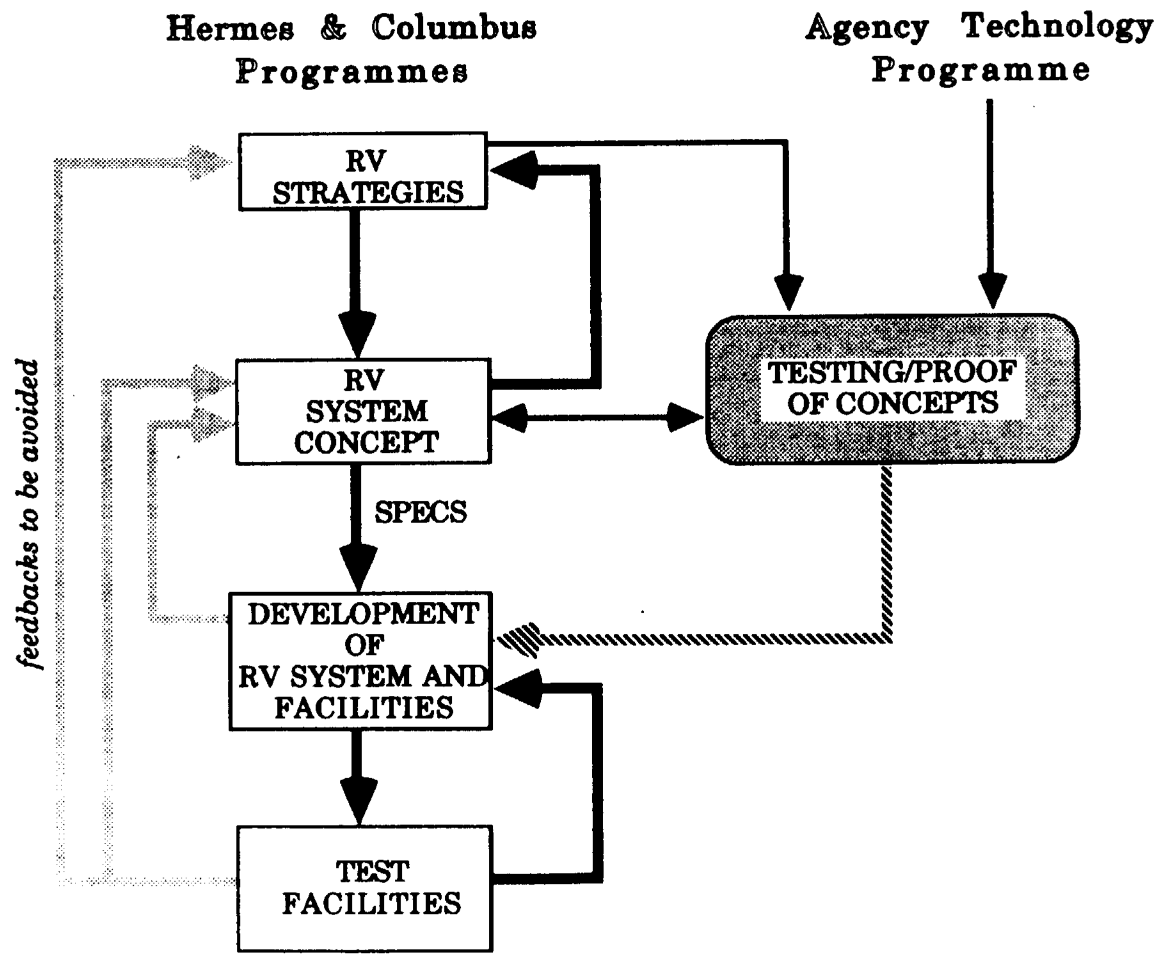
MATRA ESPACE

esa

RVD System Proof-Of-Concept in Europe

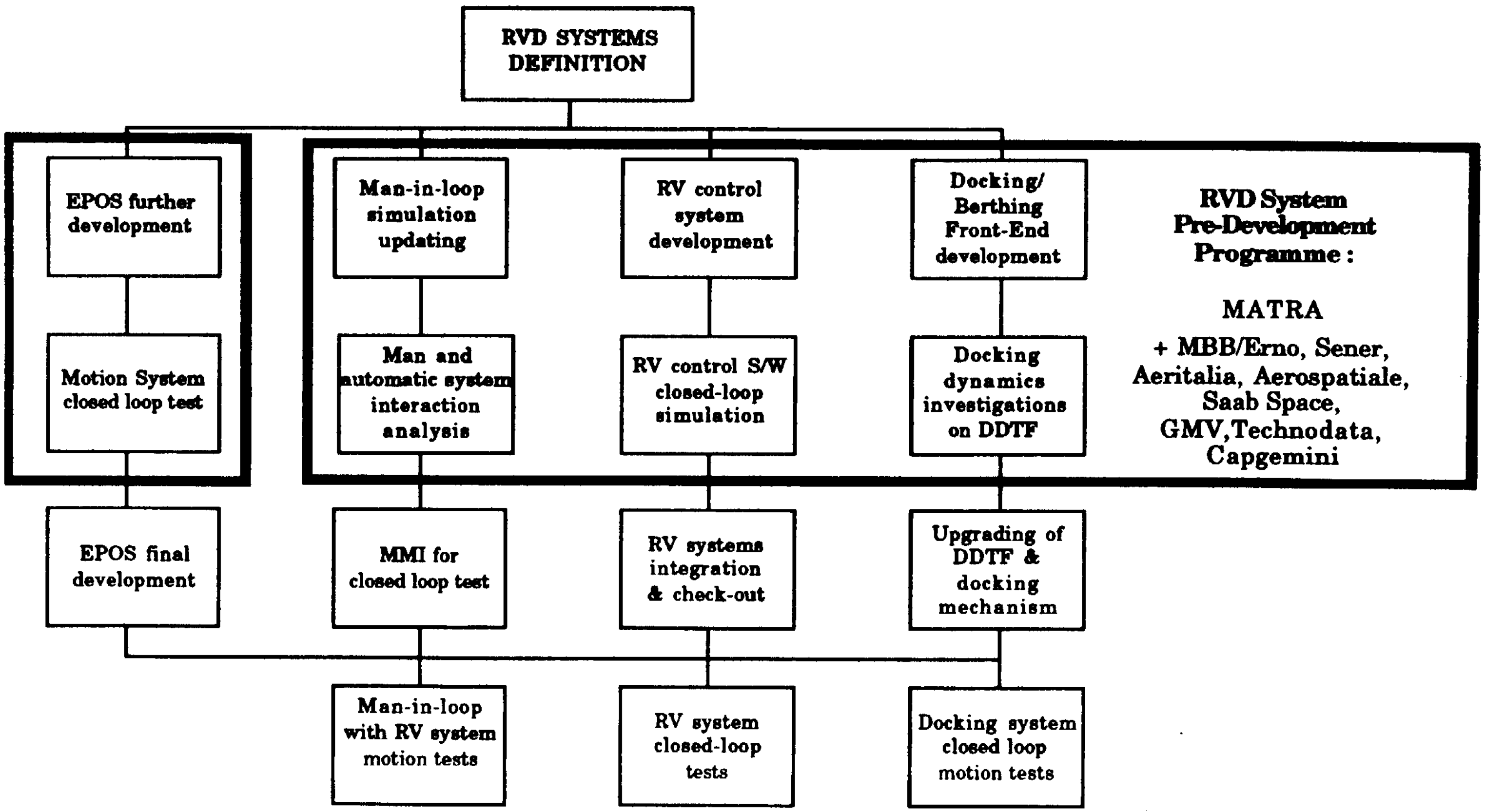
- Hermes and Columbus programmes necessitate design/development of RV system
- Numerous studies performed by European Space Agency (ESA) on RV and proxops
- Necessity to synchronize RV-related projects activity to avoid mis-design of RV system
- A RV System Proof-Of-Concept programme has been initiated by ESA
- Assessment of Hermes and Columbus Requirements and Baselines
- Setup of RV System including projects commonalities, and extended to mission/spacecraft alternatives (AR5/MIR, scenarios, equipments,...)
- Utilization of material available in Europe:
  - hardware and software developed under ESA contracts
  - test facilities developed in France (Docking Dynamics Test Facility, CNES/MATRA) and in FRG (European Proximity Operations Simulator, ESA/DLR)

RVD System Proof-Of-Concept in Europe

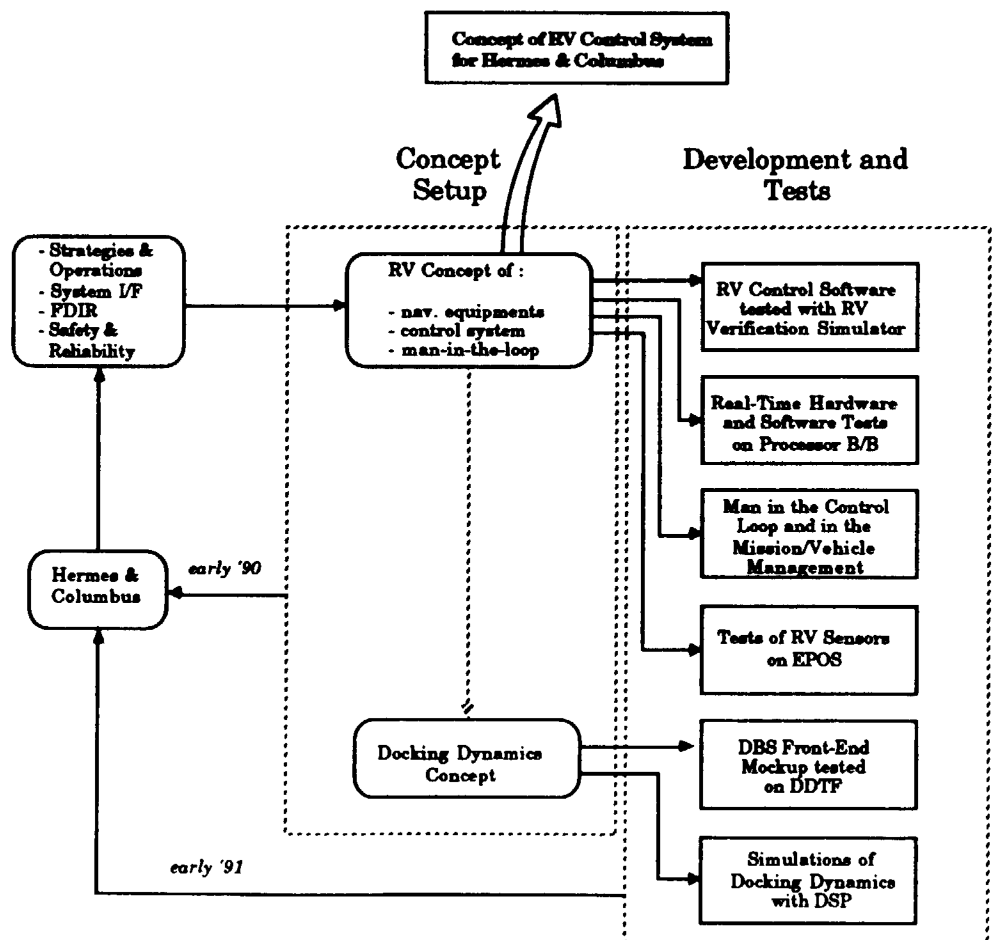


RVD System Proof-Of-Concept in Europe

SCOPE OF PROGRAMME FOR A COMPLETE PROOF-OF-CONCEPT FOR RVD

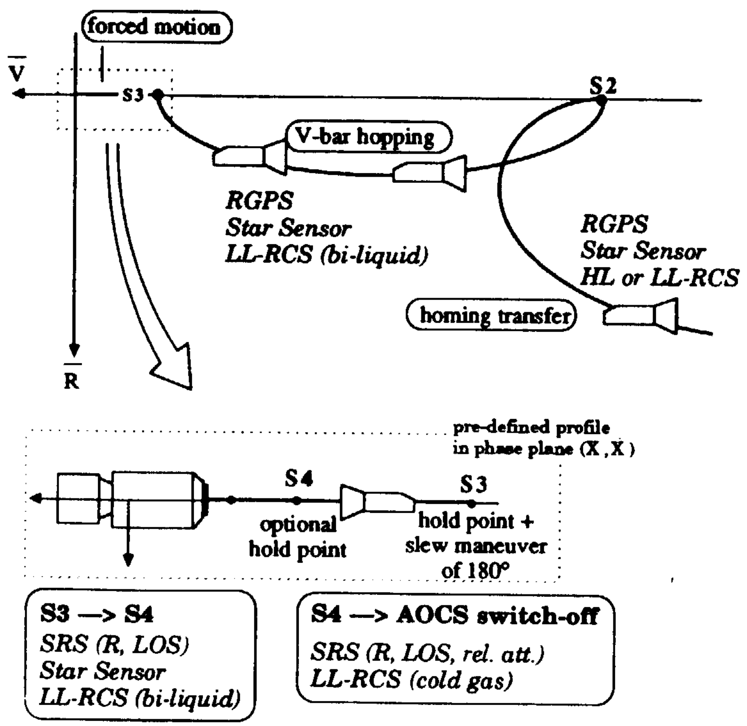


Hermes/Columbus RV Control System within European RV System Pre-Development Programme

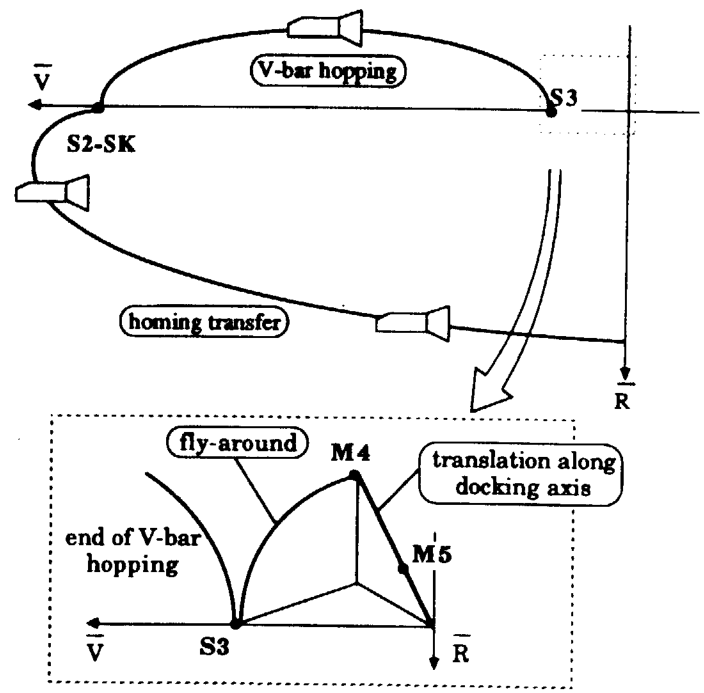


Mission Scenarios and Equipments - Operations with Columbus Free Flyer

Earth-Pointing CFF

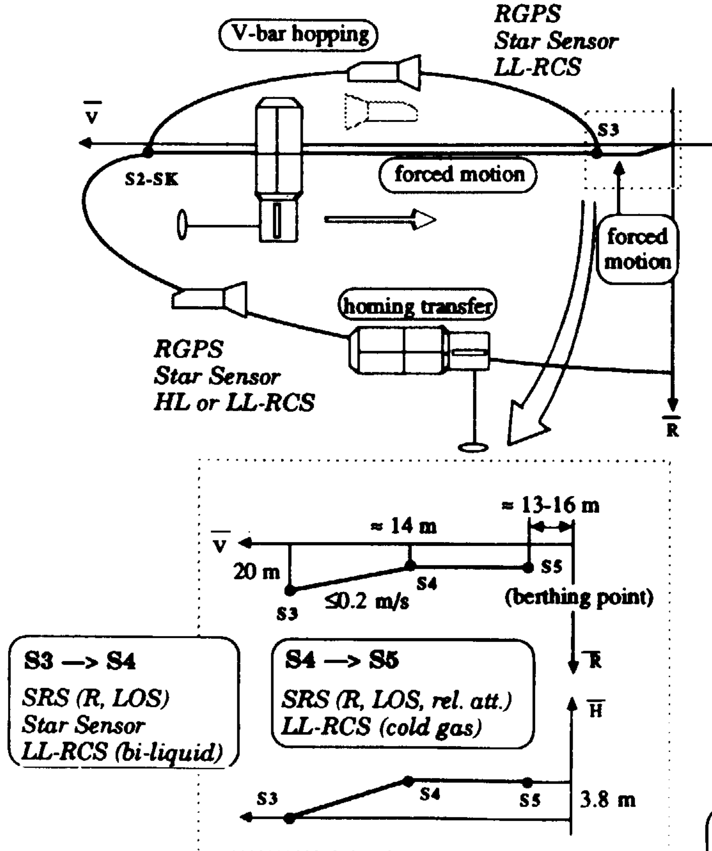


Sun-Pointing CFF

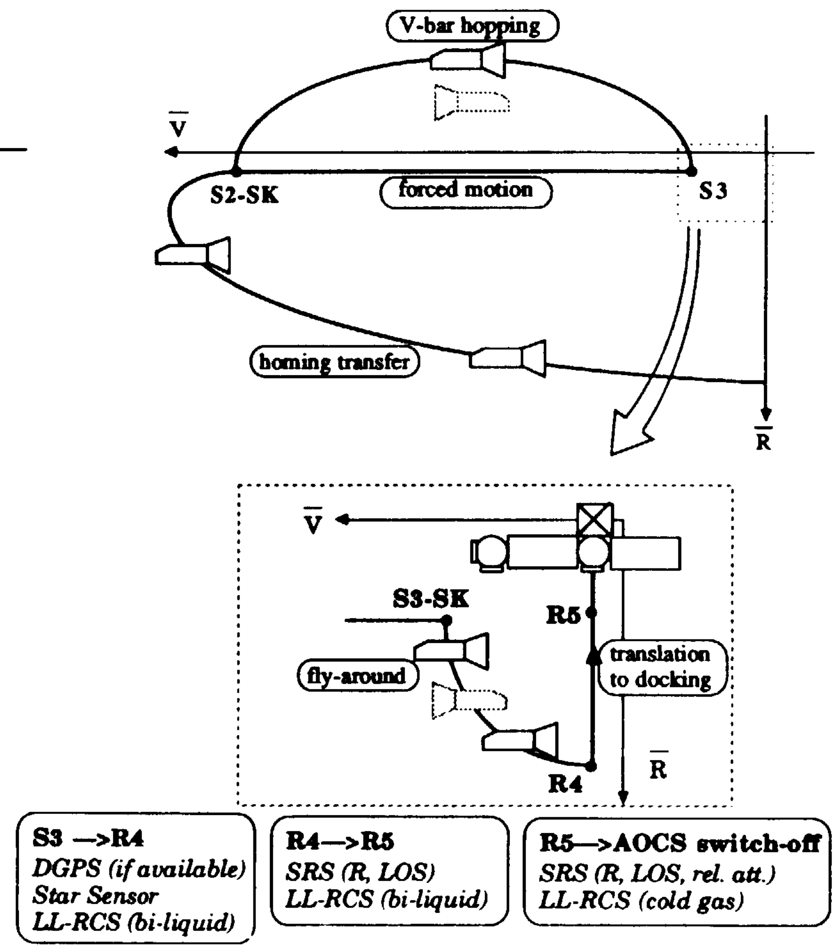


Mission Scenarios and Equipments - Operations with Space Station Freedom

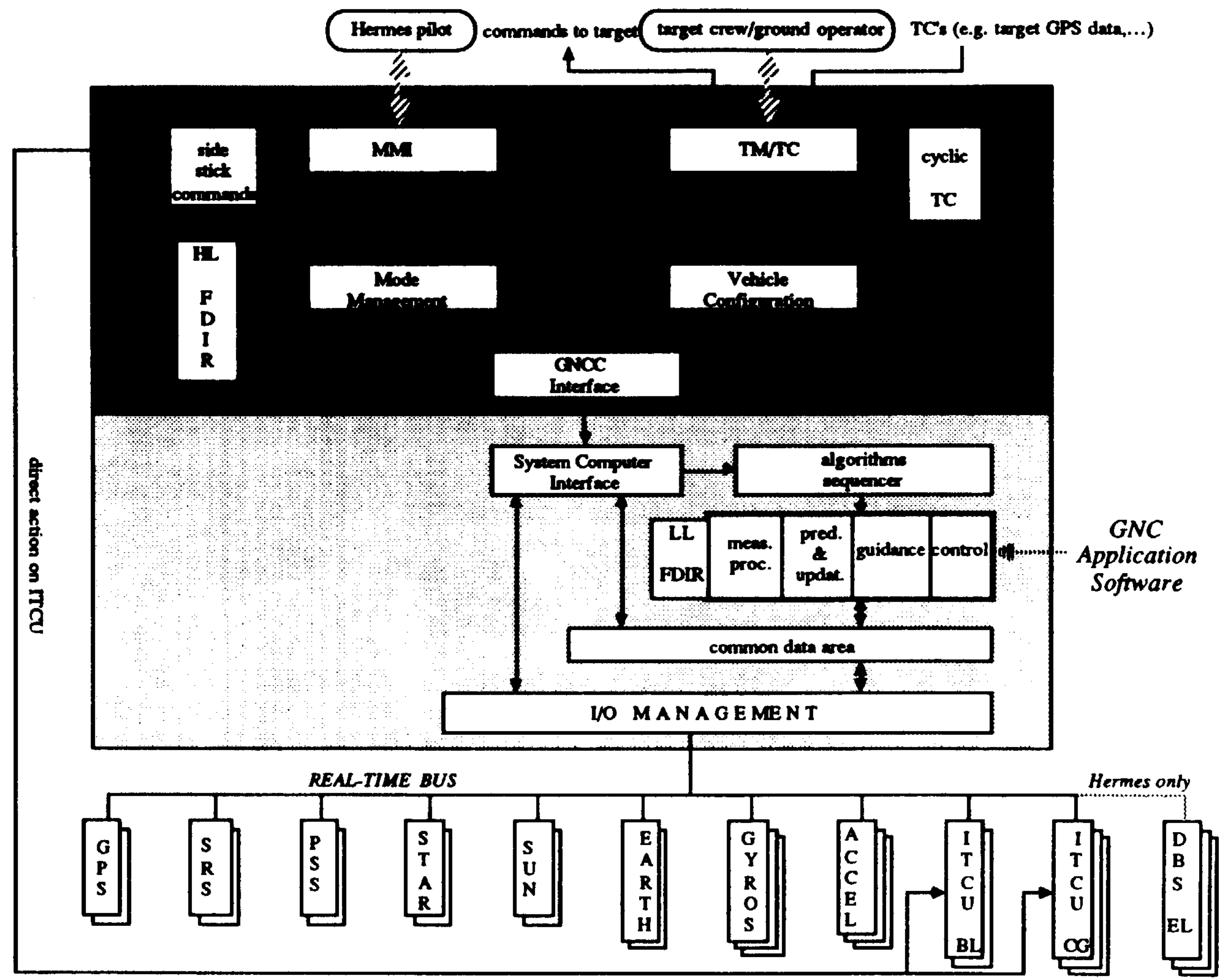
Berthing Operations



Docking Operations



Functional Description of RV Control System Concept



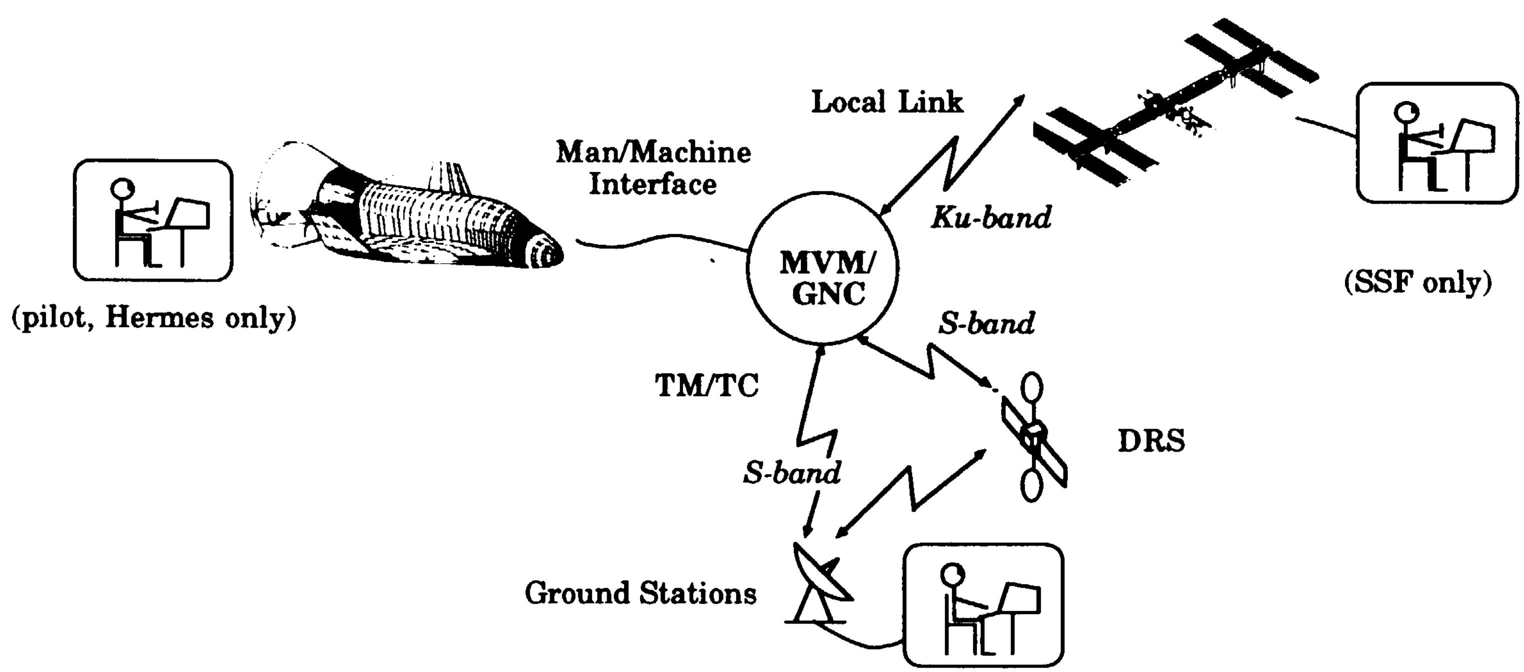
Functional Description of RV Control System Concept

- RV Control on-board software runs GNC computer and the "system" computer
- all commonalities between Hermes and Columbus + alternative solutions wherever differences are identified.
- participation of several human operators to proxops:
  - the <u>chaser pilot</u> (Hermes):
    - supervisory control during all proxops
    - active intervention in spaceplane control during the final part of the mission (state update, manual control)
  - the <u>target crew</u> (space station):
    - monitoring activities
    - "red button" capability in close chaser vicinity
  - the <u>ground operator(s)</u>:
    - monitoring and supervisory activities

Functional Description of RV Control System Concept

- Redundancy concept accounts for similar concepts for both Hermes (FO/FS) and Columbus (FO/FO/FS)
- Considered concept is FO/FS (check of 2 types of transitions upon failures):
    - transition after 1st failure (FO requirement)
    - transition after 2nd failure (FS requirement)
- 2 types of FDIR:
    - low level FDIR (failures occuring at equipment level having their own BITE)
    - high level FDIR (monitoring of the spacecraft behaviour against nominal expected values, active safety)
- RV Control on-board software:
    - Mission and Vehicle Management (MVM) functions
    - Guidance, Navigation and Control (GNC) functions

Functional Description of RV Control System Concept



Example of Guidance and Navigation during homing transfer

- ➜ specification at end of homing transfer

$\delta x < 100$ m	$\delta \dot{x} < 2$ cm/s
$\delta y < 10$ m	$\delta \dot{y} < 1$ cm/s
$\delta z < 10$ m	$\delta \dot{z} < 1$ cm/s

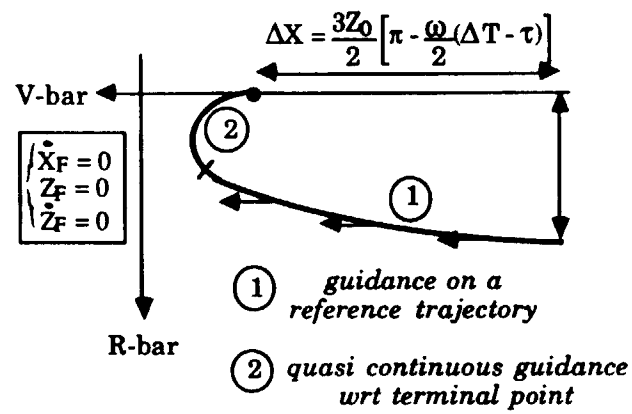
- ➜ navigation : Relative GPS
- ➜ transfer principle : (N) boosts parallel to target velocity direction (V-bar)
- ➜ parameters : number of boosts (N), boost duration (τ), time interval between boosts (ΔT)
- ➜ $N = 2 \Rightarrow$ Hohmann-type transfer

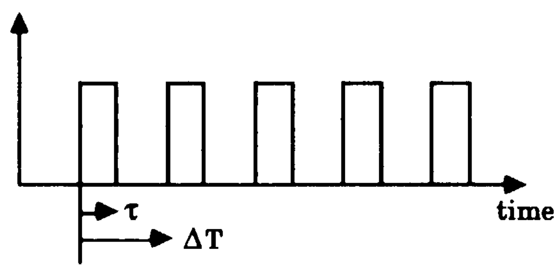
  $N \approx 100 \Rightarrow$ (quasi-)continuous thrust
  - terminal guidance (wrt aimed point)
  - guidance on trajectory

- ➜ strategy comparison (CNES Research and Technology Programme)

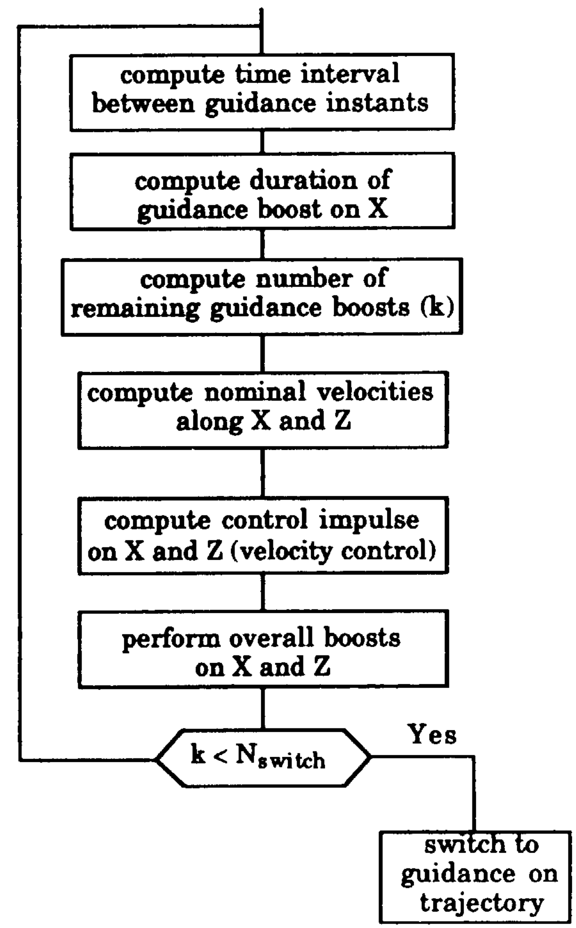
strategy	performances	robustness	fuel consumption	algorithm simplicity
Hohman transfer	—	—	+	— —
Control on trajectory (N boosts)	+	+	—	+
Terminal point guidance (N boosts)	+	++	+	—

Example of Guidance and Navigation during homing transfer



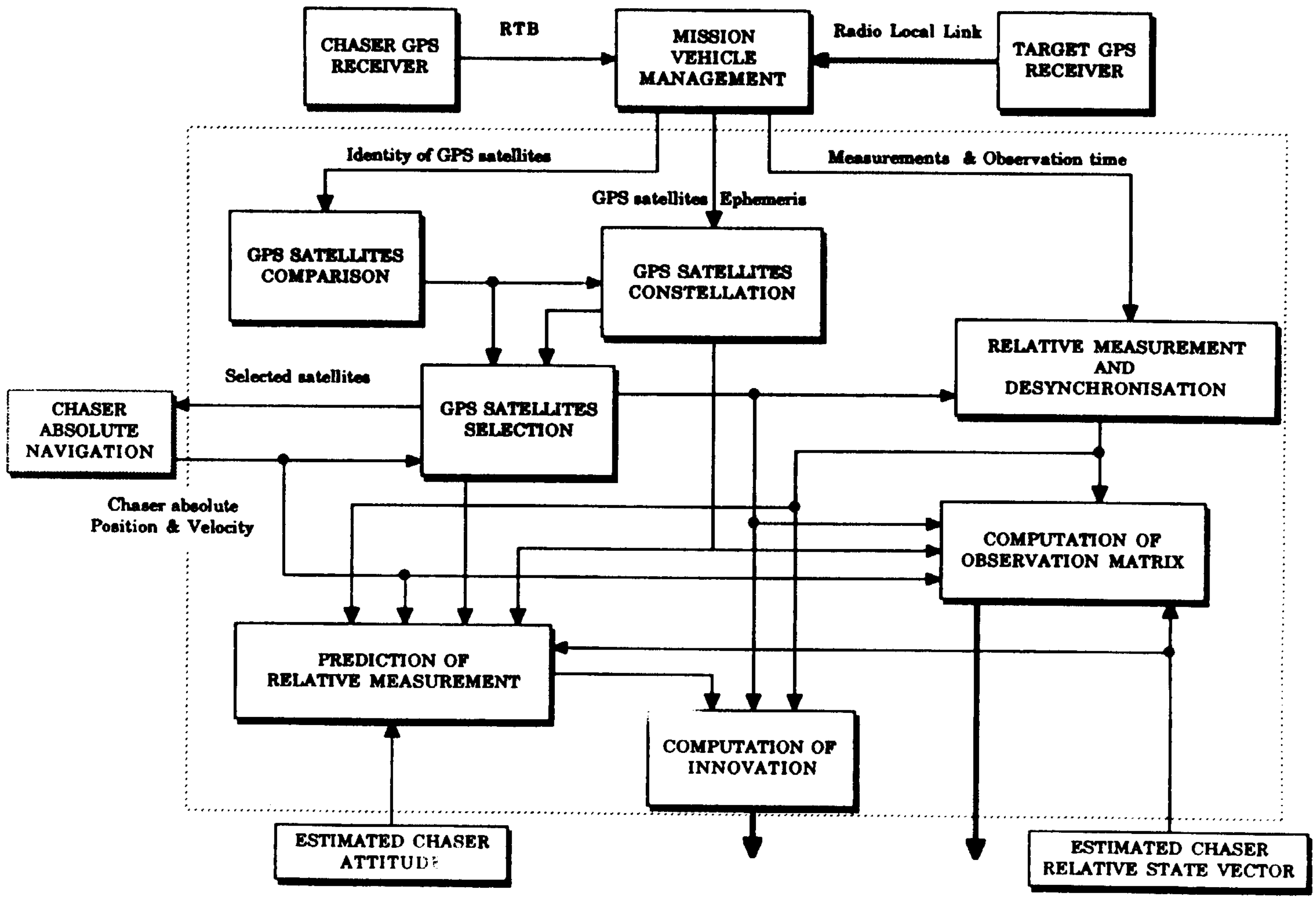


$$\begin{cases} \tau = \dfrac{\omega Z_0}{2N\gamma_{max}} \\ \Delta T = \dfrac{2\pi}{N\omega} \end{cases}$$



Example of Guidance and Navigation during homing transfer

GPS Measurement Processing

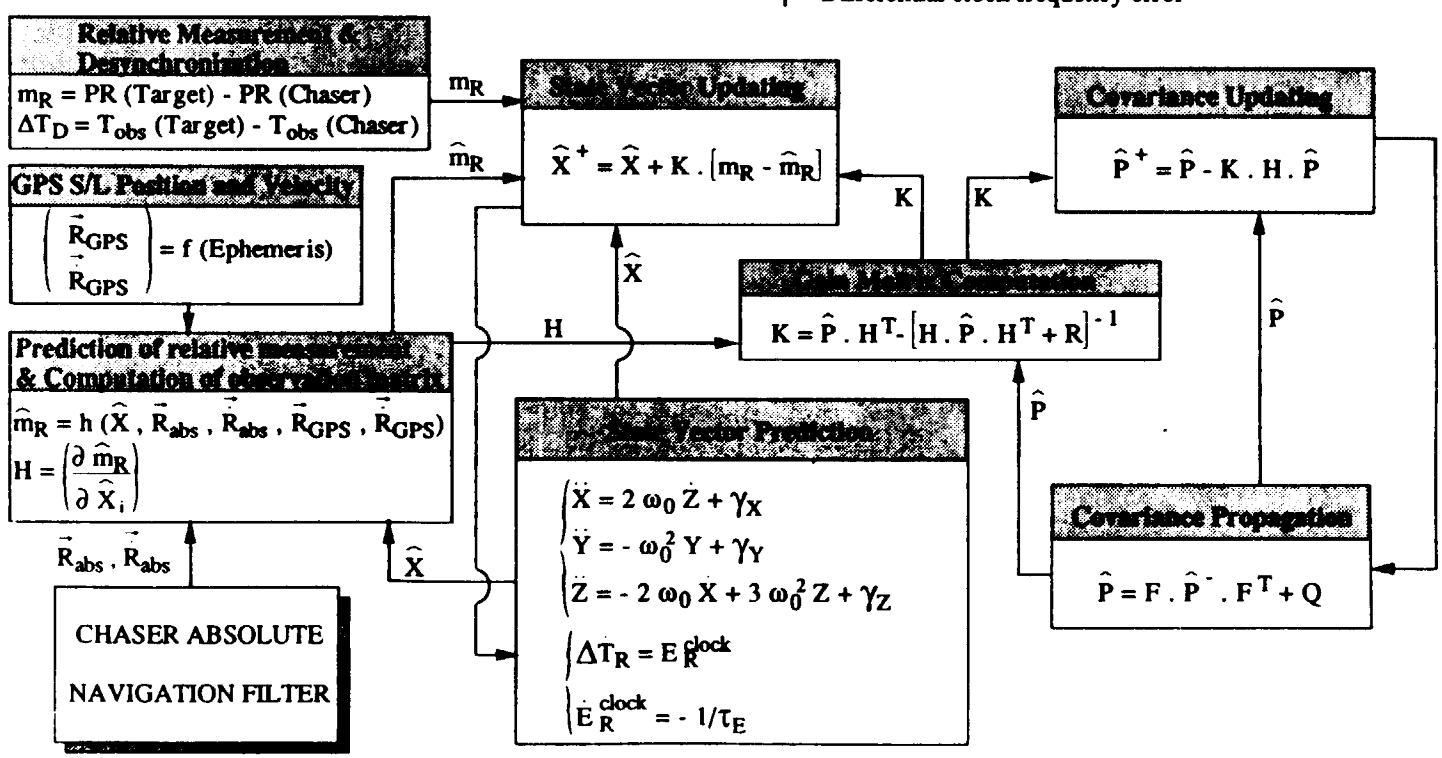


Example of Guidance and Navigation during homing transfer

GPS Relative Navigation Filter

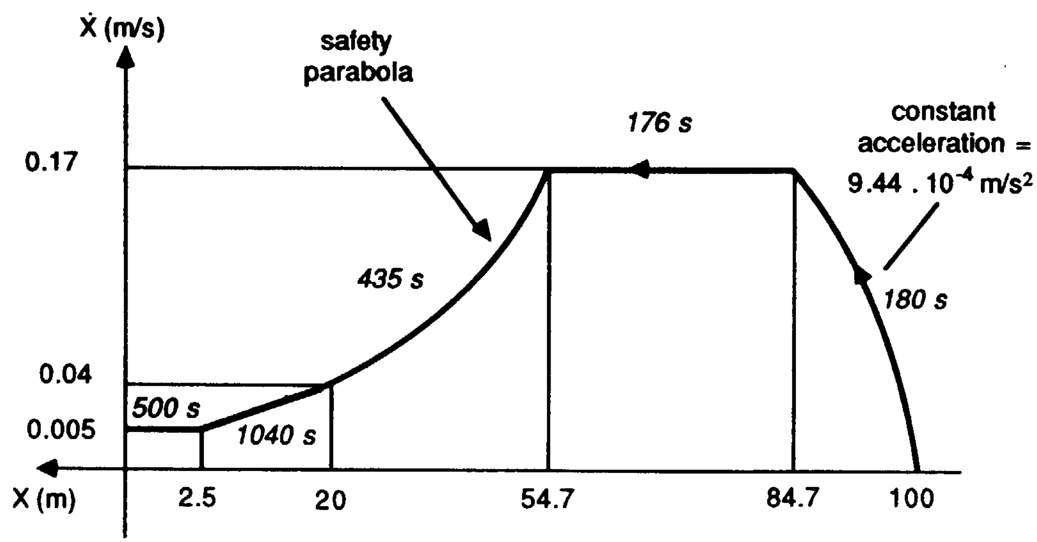
DATA PROVIDED TO GNC FOR EACH SELECTED GPS S/L :
- Pseudo-Range measured by Target receiver and observation time
- Pseudo-Range measured by Chaser receiver and observation time
- GPS satellite Ephemeris

RELATIVE NAVIGATION FILTER STATE VECTOR :
- Chaser relative Position in Target local orbital frame
- Chaser relative Velocity in Target local orbital frame
- Differential clock error
- Differential clock frequency error



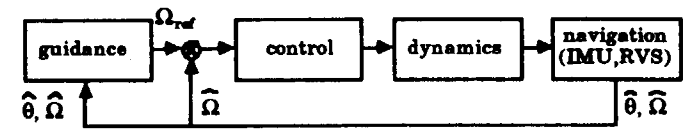
Guidance, Navigation and Control during translation to docking

- ➔ predefined profile in phase plane for approach along docking axis
  - constant acceleration
  - constant velocity
  - constant braking
  - exponential braking
  - constant velocity
- ➔ constant and exponential braking ensure passive safety
- ➔ overall duration approximately 40 minutes



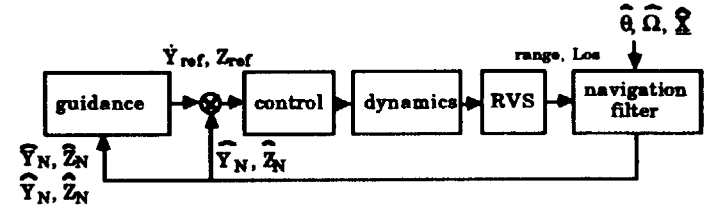
Guidance, Navigation and Control during translation to docking

→ navigation state vector : $\hat{\underline{\dot{Y}}}_N = \begin{pmatrix} \hat{\underline{X}}_{DC/DT} \\ \hat{\underline{\dot{X}}}_{SC/ST} \\ \hat{\underline{\theta}}_{DC/DT} \\ \hat{\underline{\Omega}}_{ST/GT} \end{pmatrix}$

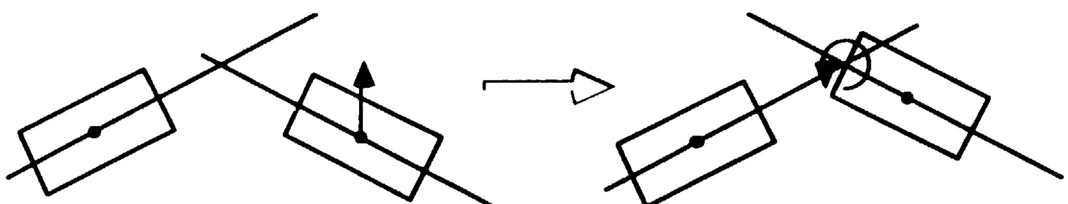


Attitude chain

→ control state vector : $\hat{\underline{\dot{Y}}}_C = \begin{pmatrix} \hat{\underline{X}}_{DC/DT} \\ \hat{\underline{\dot{X}}}_{DC/DT} \\ \hat{\underline{\theta}}_{DC/DT} \\ \hat{\underline{\Omega}}_{DC/DT} \end{pmatrix}$



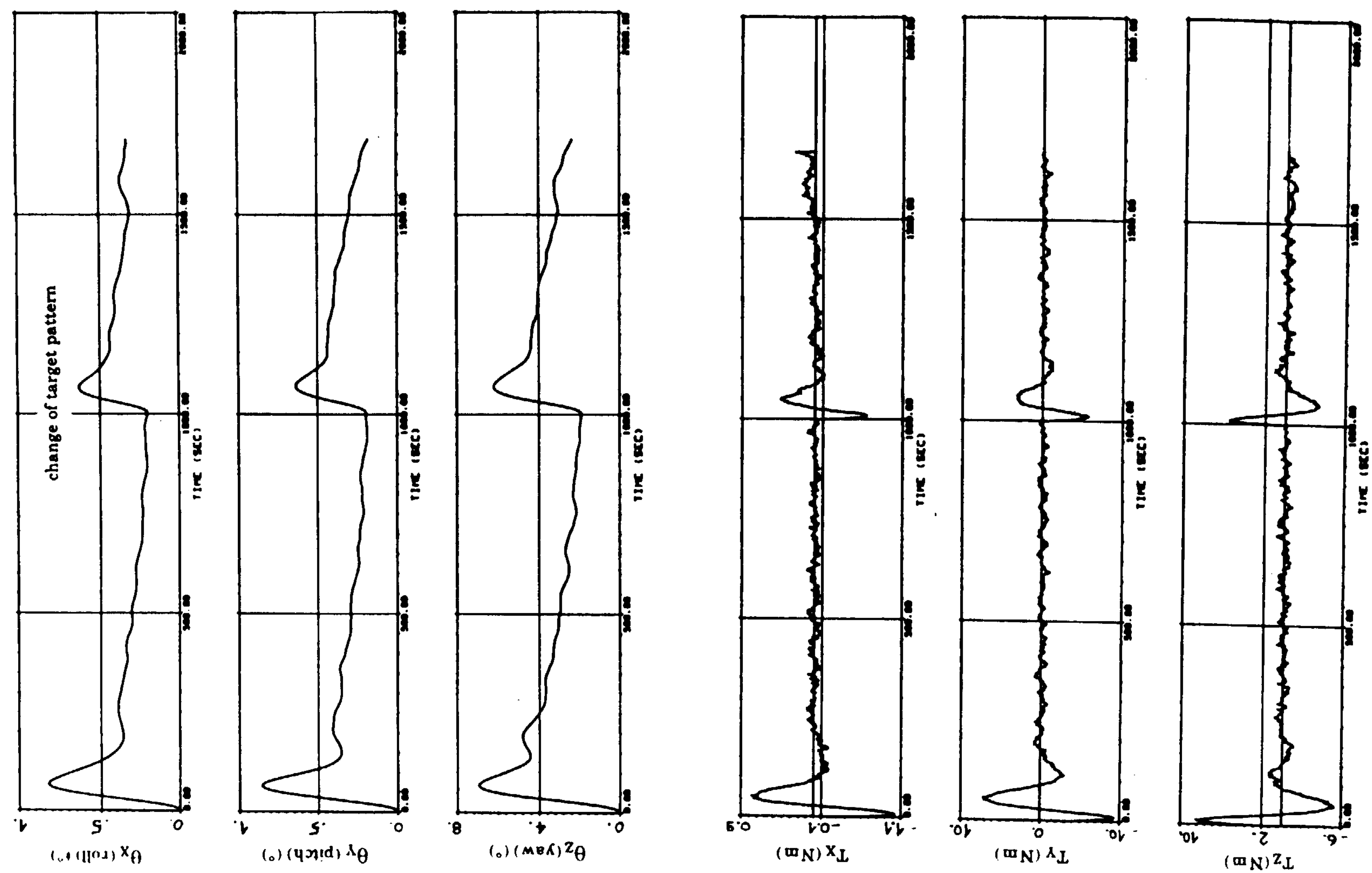
Lateral Position chain



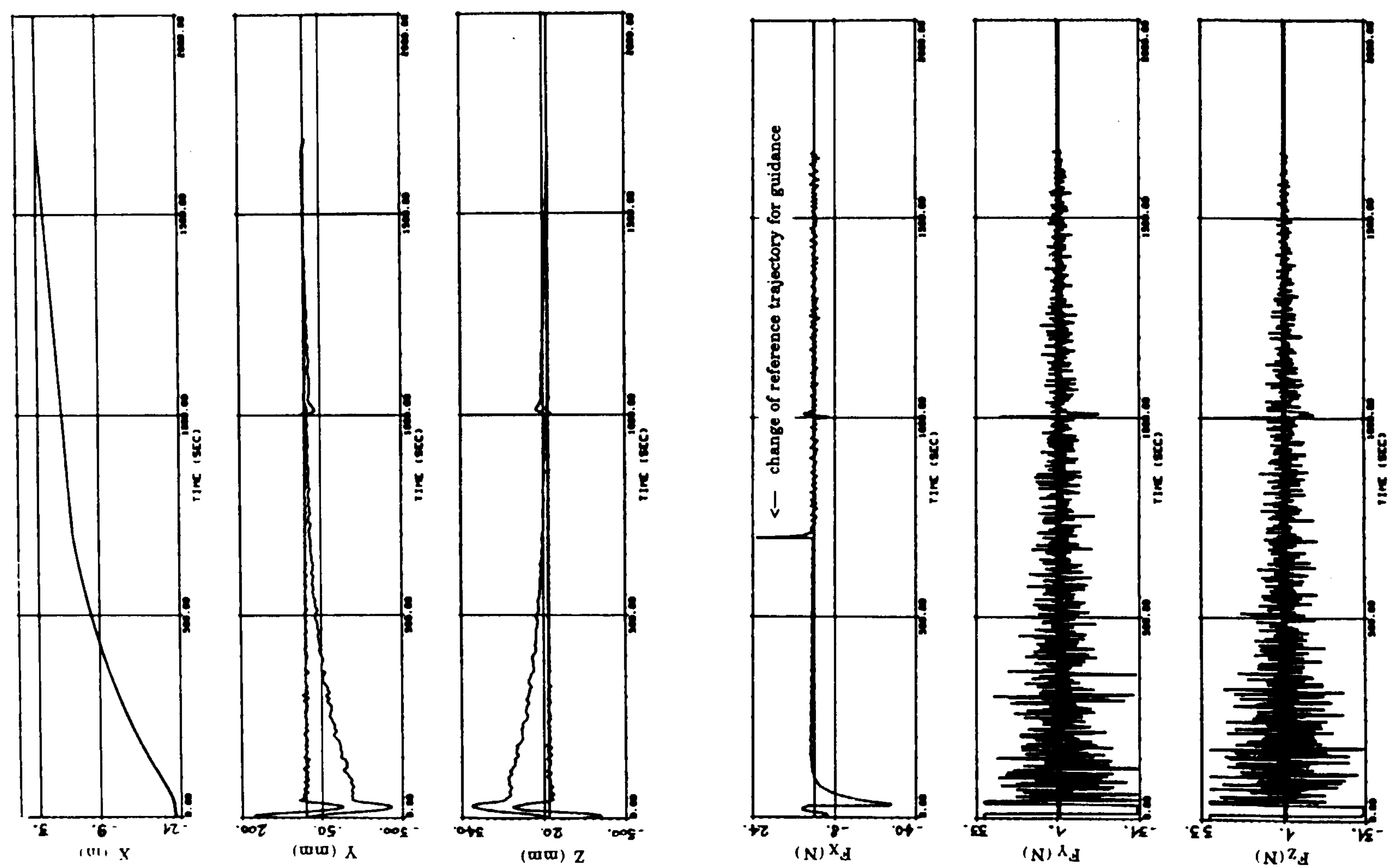
Control Principle

$$\begin{pmatrix} F_C \\ T_C \end{pmatrix} = \begin{pmatrix} * & * & * & * \\ 0 & 0 & * & * \end{pmatrix} \cdot \begin{pmatrix} z \\ \dot{z} \\ \theta_r \\ \Omega_r \end{pmatrix}$$

Guidance, Navigation and Control during translation to docking (cont'd)

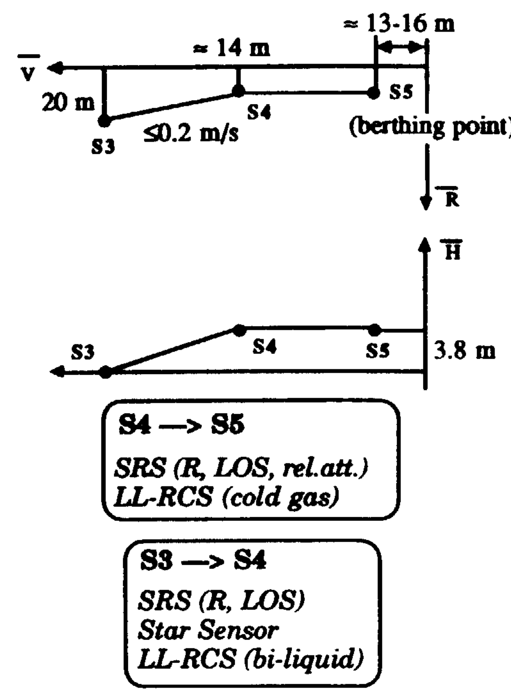


Guidance, Navigation and Control during translation to docking (cont'd)



Guidance, Navigation and Control during translation to berthing

- mission of CFF to SSF
- estimation/control of CFF position wrt SSF body frame
  estimation/control of CFF attitude wrt LVLH frame
- "chaser to remain 3-5 minutes within a berthing box (2m x 2m x 2m)"
- use of short range sensor
  ($\Delta R < 1\%$ of R; $\Delta LOS \leq 0.01°$)
  attitude prediction based on gyros
  short range sensor to be tilted by 15 degrees
  SSF provides information on its attitude/attitude rate



Guidance, Navigation and Control during translation to berthing (cont'd)

➜ error budget at S5

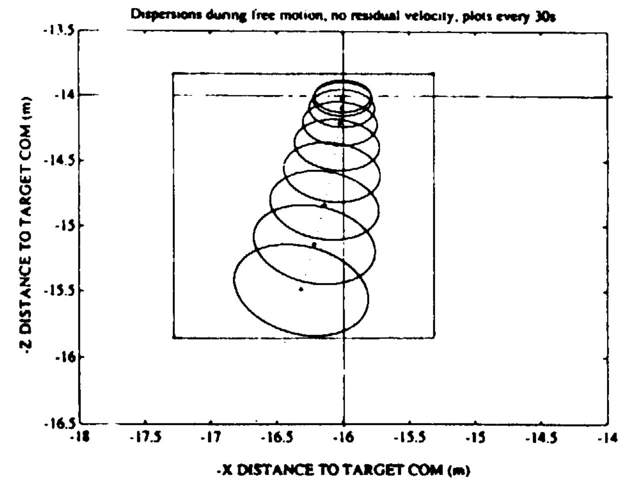
	axial position (cm)	axial velocity (mm/s)	lateral position (cm)	lateral velocity (mm/s)
global performance	21.5	0.7	12	0.2
range bias	17.3	-	6	-
LOS bias	0.08	-	0.3	-
range noise	1.3	0.5	0.4	0.12
LOS noise	-	-	-	-
target motion	2.9	-	5.5	0.05
dirturbances	0.17	0.15	-	0.01

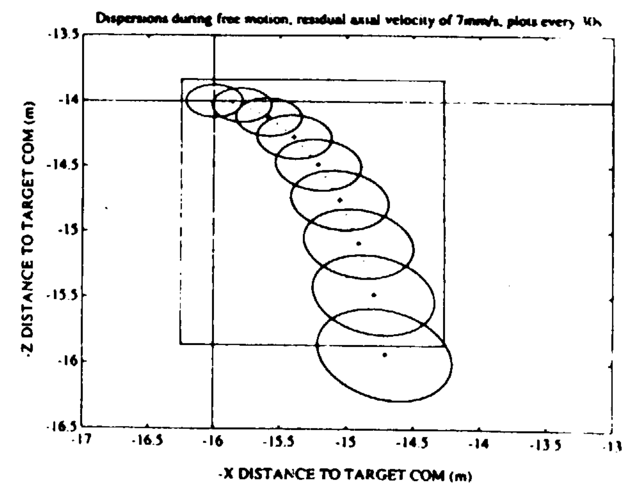
➜ specification on dispersions at S5

axial position	25 cm
axial velocity	2 mm/s
lateral position	15 cm
lateral velocity	1 mm/s
angular velocity	0.01 °/s

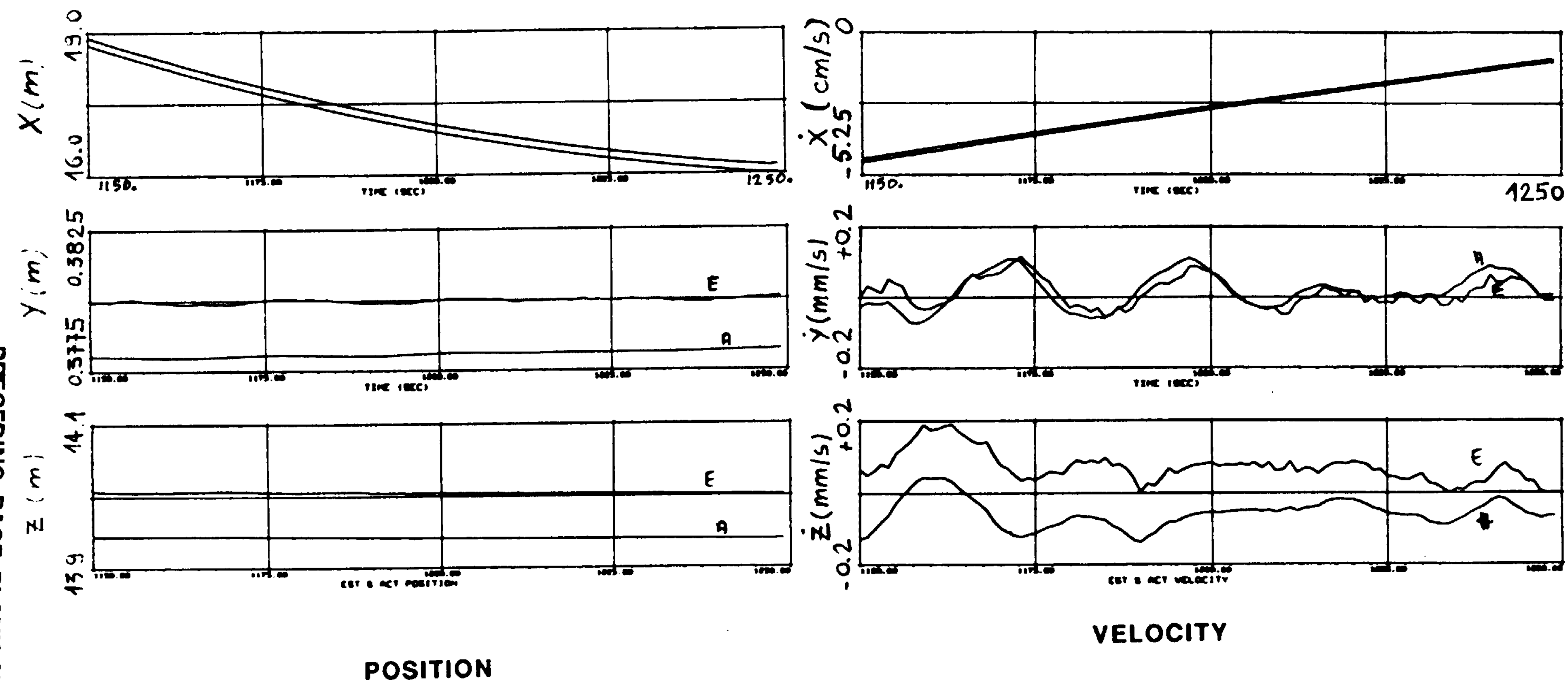
➜ residual axial velocity

$0 \rightarrow 7$ mm/s





Guidance, Navigation and Control during translation to berthing (cont'd)



PRECEDING PAGE BLANK NOT FILMED

PAGE 154 INTENTIONALLY BLANK

Non Real-Time Verification Testing

➔ Performance of non-real time software tests on a mainframe computer

➔ Development of RV Control Software covering all mission phases/modes

- Hermes, Columbus Free Flyer, Space Station Freedom
- nominal approaches, contingency operations, retreat maneuvers

➔ Development of RV Verification Software (RV Control Software "environment")

- dynamics modelling (rigid + flexible spacecraft)
- equipment modelling (sensors, actuators)
- perturbations modelling (plume impingement, air drag, liquid sloshing,...)
- DHS Simulation
- Interface with operators (mission operators, simulation operator)

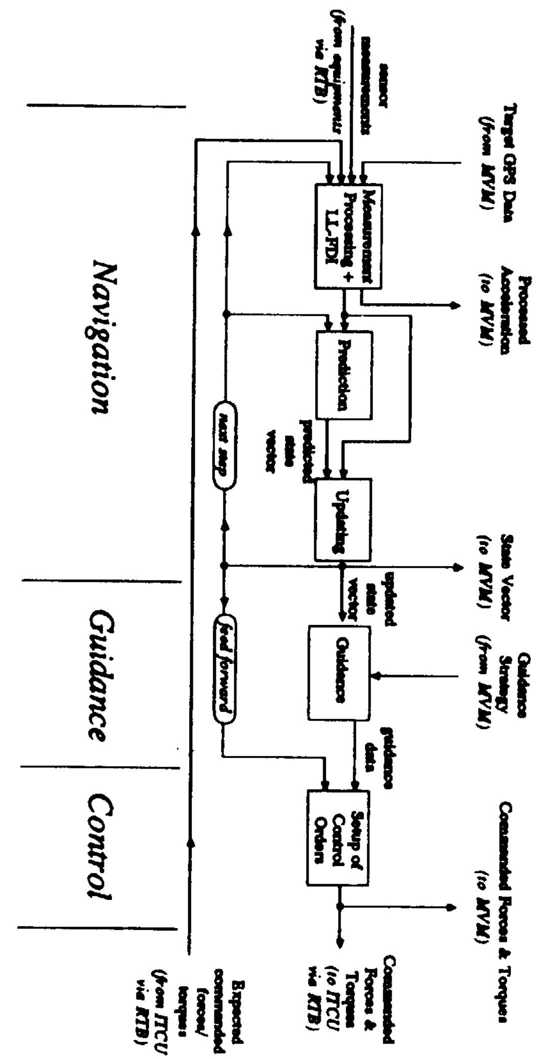
➔ RV Control and Verification Software developed in C-language

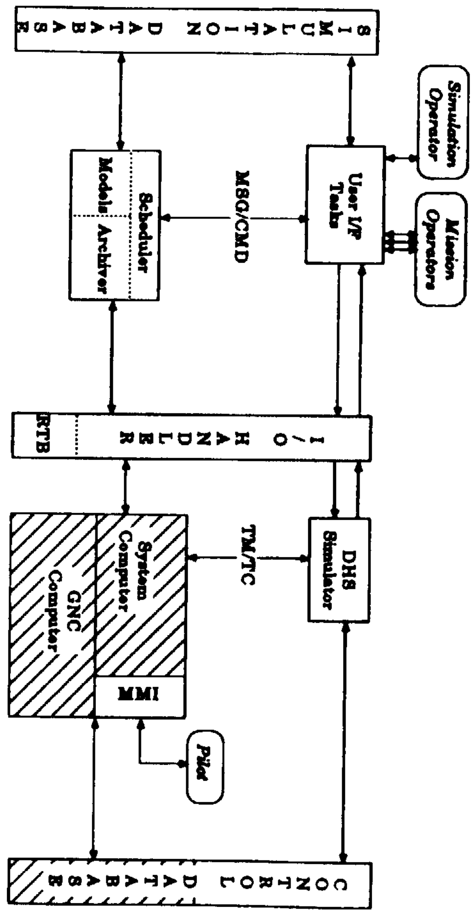
➔ Tests aiming at verifying suitability of RV Control System:

- in terms of trajectories, equipments, GNC, FDIR
- for mission/vehicle management
- for intervention of operators (pilot, target, ground)

➔ Non real-time software tests $\Rightarrow$ no state update/manual control verification

Non Real-Time Verification Testing- Simplified Software Architecture



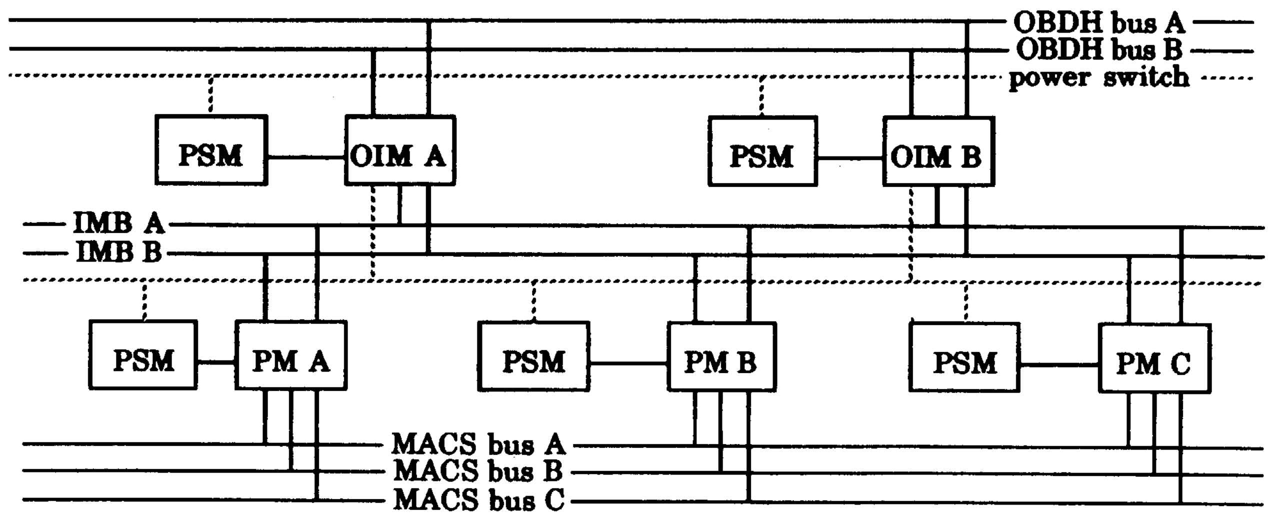


PAGE 158 INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

Real-Time Verification Testing

- Use of a breadboard of GNC computer developed by ESA
- Two failure tolerant architecture (3 processing modules, 3 real-time buses for equipment I/F
- Fail stop concept
- Interface with DHS through dual OBDH bus system
- Interface with equipments through triple MACS bus system



Real-Time Verification Testing

- ➜ Application software developed in ADA (TLD compiler)

- ➜ Development of environment software allowing real-time I/O and synchronization (developed in C-language)

- ➜ Tests aiming at verifying:

  - impact of S/W architecture on real-time performance
  - impact of ADA on real-time performance
  - suitability of FO/FS concept to missions wrt FDIR strategy

- ➜ Verification tests performed for a highly demanding GNC mode (final translation prior to direct docking)

➔ References

1 C. Pauvert (1990)
Docking Dynamics : Analysis of Front-End Requirements
Autonomous RVD Conference, Houston (USA)

2 C. Pauvert (1990)
Description and Performances of the MATRA CCD Camera Sensor
Autonomous RVD Conference, Houston (USA)

3 M. Le Du (1990)
Overview of CNES Rendezvous and Docking Activities
Autonomous RVD Conference, Houston (USA)

Activities reported in this presentation have been performed in the frame of the Technology Programme of the European Space Agency (ESA) and of the Technology Programme of the French Space Agency (CNES)

☆ U.S. GOVERNMENT PRINTING OFFICE: 1990—761-014/20348